Universität Bielefeld

Technische Fakultät
Abteilung Informationstechnik
Forschungsberichte

Sequencing from compomers: Using mass spectrometry for DNA de-novo sequencing of 200+ nt

Sebastian Böcker

Report 2003-04

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

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

ISSN 0946-7831

SEQUENCING FROM COMPOMERS: USING MASS SPECTROMETRY FOR DNA DE-NOVO SEQUENCING OF 200+ NT

SEBASTIAN BÖCKER

Abstract. One of the main endeavors in today's Life Science remains the efficient sequencing of long DNA molecules. Today, most de-novo sequencing of DNA is still performed using electrophoresis-based Sanger Sequencing, based on the Sanger concept of 1977. Methods using mass spectrometry to acquire the Sanger Sequencing data are limited by short sequencing lengths of 15–25 nt.

We propose a new method for DNA sequencing using *base-specific cleavage* and *mass spectrometry*, that appears to be a promising alternative to classical DNA sequencing approaches. A single stranded DNA or RNA molecule is cleaved by a base-specific (bio-)chemical reaction using, for example, RNAses. The cleavage reaction is modified such that not all, but only a certain percentage of those bases are cleaved. The resulting mixture of fragments is then analyzed using MALDI-TOF mass spectrometry, whereby we acquire the molecular masses of fragments. For every peak in the mass spectrum, we calculate those base compositions that will potentially create a peak of the observed mass and, repeating the cleavage reaction for all four bases, finally try to uniquely reconstruct the underlying sequence from these observed spectra. This leads us to the combinatorial problem of Sequencing From Compomers and, finally, to the graph-theoretical problem of finding a walk in a subgraph of the de Bruijn graph. Application of this method to simulated data indicates that it might be capable of sequencing DNA molecules with 200+ nt.

1. Introduction

Suppose we want to reconstruct an (unknown) string s over the alphabet Σ. Multiple copies of s are cleaved with a certain probability whenever a specific character $x \in \Sigma$ appears, comparable to the Partial Digestion Problem (Waterman, 1995). Then, every resulting fragment y is scrambled by a random permutation so that the only information we are left with is how many times y contains each character $\sigma \in \Sigma$. In addition, we discard all fragments y that contain the cleavage character more than k times for a fixed threshold k. This threshold is usually chosen very small, for example $k \in \{2, 3, 4\}$. If we are given such reduced and scrambled fragment sets for every character $x \in \Sigma$, can we uniquely reconstruct the string s from this information? The main challenge for such reconstruction is not scrambling the fragments, but discarding fragments containing too many cleavage characters. Nevertheless, it is often possible to reconstruct the string.

The above problem arises in the context of sequencing DNA by the use of mass spectrometry. Today, most de-novo sequencing of DNA without any *a priori* information regarding the amplicon sequence under examination, is still performed based on the Sanger concept, see Sanger et al. (1977). Maxam and Gilbert (1977) proposed a method utilizing base-specific chemical cleavage, but this method has not been viable for the dramatically increased demand in DNA sequencing. Both sequencing technologies use gel or capillary electrophoresis to acquire the experimental data. Other approaches like combining the Sanger concept with mass spectrometry for data acquisition (Köster et al., 1996), or PyroSequencing (Ronaghi et al., 1998) are limited by the short sequencing length of 15–25 nt, while Sequencing by Hybridization (SBH) (Bains and Smith, 1988; Drmanac et al., 1989; Lysov et al., 1988) never

Date: June 18, 2003.

Sebastian Böcker is currently supported by "Deutsche Forschungsgemeinschaft" (BO 1910/1-1) within the Computer Science Action Program. This work was carried out in part while Sebastian Böcker was employed by SEQUENOM GmbH, Hamburg, Germany.

became practical due to the high number of false reads as well as the current costs of SBH chips.

Here we propose a new approach to DNA sequencing that is *not* based on the Sanger concept, using MALDI-TOF mass spectrometry to acquire the experimental data. Since MALDI-TOF mass spectrometry reads can be obtained in milliseconds to seconds, compared to hours for electrophoresis reads, and mass spectrometry generally provides reliable and reproducible results even under high throughput conditions, our approach seems to be a promising alternative to traditional electrophoresis-based de-novo sequencing. We have applied our method to simulated mass spectra generated from random as well as biological sequences, and simulation results indicate high chances of successful reconstruction even when sequencing 200 and more nucleotides. The reconstruction accuracy, however, highly depends on the underlying sample sequence.

The main focus of this paper is to give a suitable mathematical formulation for the problem of reconstructing the sample sequence from compomers — that represent the randomly scrambled fragments — and to propose a branch-and-bound algorithm that is usually sufficient to reconstruct the sequence in reasonable runtime.

2. Experimental setup and data acquisition

Suppose we are given a target DNA molecule (or *sample DNA*) of length 100–500 nt. Using polymerase chain reaction (PCR) or other amplification methods we amplify the sample DNA. We assume that we have a way of generating a single stranded target, either by transcription or other methods,[1] and we talk about sample DNA even though the cleavage reaction might force us to transcribe the sample to RNA. We cleave the single stranded sequence with a base-specific chemical or biochemical cleavage reaction: Such reactions cleave the amplicon sequence at exactly those positions where a specific base can be found. Such base-specific cleavage can be achieved using endonucleases RNAse A (Rodi et al., 2002) and RNAse T1 (Hartmer et al., 2003), uracil-DNA-glycosylase (UDG, see von Wintzingerode et al., 2002), pn-bond cleavage (Shchepinov et al., 2001), and others.

We modify the cleavage reaction by offering a mixture of cleavable versus non-cleavable "cut bases," such that not all cut bases but only a certain percentage of them will be cleaved. The resulting mixture contains in principle all fragments that can be obtained from the sample DNA by removing two cut bases, cf. Fig. 1 for an example. We call such cleavage reactions *partial*.

MALDI (matrix assisted laser desorption ionization) TOF (time-of-flight) mass spectrometry (MS for short) is then applied to the products of the cleavage reaction, resulting in a sample spectrum that correlates mass and signal intensity of sample particles (Karas and Hillenkamp, 1988).[2] The sample spectrum is analyzed to extract a list of signal peaks with masses and intensities.

We can repeat the above analysis steps using cleavage reactions specific to all four bases — alternatively, we can apply two suitably chosen cleavage reactions twice, to forward and reverse strands. So, we obtain up to four mass spectra, each corresponding to a base-specific cleavage reaction. We repeat the following steps of the analysis for every cleavage reaction.

If the sample sequence is known, then exact chemical results of the used cleavage reactions and, in particular, the masses of all resulting fragments are known in advance and can be simulated by an *in silico* experiment. Clearly, this holds up to a certain extent only, and

[1]The method can easily be extended to deal with double stranded data, but we will concentrate in the following on single stranded data.

[2]More precisely, MALDI-TOF mass spectrometers measure "mass per charge" instead of "mass" of sample particles. To simplify matters, we speak of "mass" instead of "mass per charge" because most particles in a MALDI mass spectrum will be single charged. Even more precisely, MALDI-TOF MS does not provide us with masses but only with time-of-flight of sample particles, so calibration (correlation of time-of-flight and mass) has to be determined beforehand.

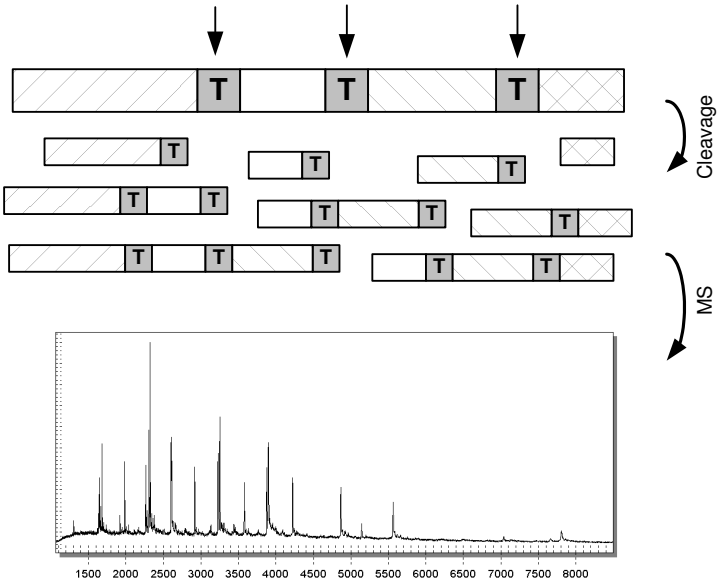

FIGURE 1. Partial cleavage using RNAse A with dCTP, rUTP, and dTTP.

measured spectra often differ significantly from the *in silico* predicted spectrum. Compared to other mass spectrometry applications, though, there is only a comparatively small number of differences between the simulated spectrum and the measured one.

Having said that, we can also solve the inverse problem: For every peak detected in the measured mass spectrum, we can calculate one or more base compositions (that is, DNA molecules with unknown order but known multiplicity of bases) that could have created the detected peak, taking into account the inaccuracy of the mass spectrometry read. Therefore, we obtain a list of base compositions and their intensities, depending on the sample DNA *and* the incorporated cleavage method. We want to stress that this calculation is simple when we are dealing with DNA or RNA, because the alphabet size is small and the average mass of a base (about 300 Dalton[3]) is much higher than the maximal mass difference between any two bases (about 50 Da). A simple algorithm based on searching $X + Y$ (Cosnard et al., 1990) can compute all base compositions in time $O(m^3)$, where m is the mass of the detected peak. Every base composition with mass sufficiently close to the detected peak must be seen as a potential explanation of the peak, and we use all such base compositions independently in the following.

Clearly, we cannot use the trivial approach of de-novo sequencing because of the exponential number of sequences: In this approach, we would (a) simulate the mass spectra for every potential sequence, for example $s \in \Sigma^l$ for some given length l, and (b) compare the resulting simulated spectra against the measured mass spectrum, finding the one that gives a best fit of the measured spectrum. Here a sequencing length of 200 nt results in about $2.6 \cdot 10^{120}$ mass spectra that we have to test for every cleavage reaction.

2.1. **Limitations.** The experimental setup described above has been successfully applied to problems such as Pathogen Identification (von Wintzingerode et al., 2002) or SNP discovery (Böcker, 2003; Rodi et al., 2002). There, information on the sample sequence(s) under consideration is known beforehand, so that the requirements to the mass spectrometer (with regards to calibration accuracy and resolution) are comparatively small. Furthermore, we can use the additional information provided by the known reference sequence to reduce the algorithmic complexity of answering such questions. In the setting of this paper, though, almost no information but the mass spectrometry data itself is available.

[3]Dalton (Da), a unit of mass equal to $\frac{1}{12}$ the mass of a carbon-12 nucleus, about 0.992 times the mass of a single H atom.

In real life, several limitations characteristic for mass spectrometry and the experimental setup make the problem of de-novo sequencing from mass spectrometry data more challenging:

(i) Current mass spectrometers limit the mass range in which particles can be detected: Signals above 8 000 Dalton (≈ 25 nt) tend to get lost in the spectrum.
(ii) Using MS, we can determine masses up to some inaccuracy only. Novel MS technologies like OTOF (orthogonal time of flight) MS allow us to measure particle masses with an inaccuracy of less than 0.3 Da, while current (ATOF, axial time of flight) mass spectrometers can show an inaccuracy of 1-2 Da under high throughput conditions.
(iii) Because MS spectra are noisy, it is often impossible to distinguish between signal peaks with low intensities and noise peaks randomly found in the spectrum. Henceforth, one has to choose between minimizing either the number of false positive, or the number of false negative detected peaks.
(iv) For a fixed cleavage reaction, several potential base compositions can have nearly identical masses. In the following, we independently use every potential explanation of a mass signal as a base composition. Therefore we transform a single mass signal found in the mass spectrum, into a list of base compositions with masses sufficiently close to the signal mass, depending on the sample sequence and the incorporated cleavage method.
(v) Using partial cleavage results in an *exponential decay* (in the number of uncleaved cut bases) of signal intensities in the mass spectrum, so peaks from fragments containing many uncleaved cut bases will be difficult or impossible to detect.
(vi) We often know 3–20 terminal bases of the sample string in advance. This can be due to primer or promoter regions used for amplification, transcription, or the like. Furthermore, the masses of terminal fragments located at beginning or end of the sample sequence in general differ from those of non-terminal fragments, and it is often possible to uniquely identify such fragments via their masses.

Depending on the underlying model of fragment ionization, we can calculate useful ratios of cleaved vs. uncleaved cut bases: Let $r \in [0,1]$ denote the portion of cleaved cut bases and $(1-r)$ the portion of uncleaved cut bases, so that the ratio equals $r : (1-r)$. Useful choices for r are $r = \frac{2}{3}$, $\frac{1}{2}$, and $\frac{1}{3}$, because these choices maximize peak intensities of certain types of fragments in the mass spectrum. In addition, the use of small $r \ll \frac{1}{2}$ is not recommended because then, it becomes difficult to discriminate between so-called noise peaks and *any* type of signal peaks in the mass spectrum.

3. Methods

3.1. **The compomer spectrum.** Let $s = s_1 \dots s_n$ be a string over the alphabet Σ where $|s| = n$ denotes the *length* of s. We denote the concatenation of strings a, b by ab, the empty string of length 0 by ϵ.

If $s = axb$ holds for some strings a, x, b then x is called a *substring* of s, a is called a *prefix* of s, and b is called an *suffix* of s. We define the *number of occurrences* of x in s by:

$$\mathrm{ord}_x(s) := \max\big\{k \,:\, \text{there exist } s_0, \dots, s_k \in \Sigma^* \text{ with } s = s_0 x s_1 x \dots x s_k\big\}$$

Hence, x is a substring of s if and only if $\mathrm{ord}_x(s) \geq 1$. For $x \in \Sigma^1$, $\mathrm{ord}_x(s)$ simply counts the number of appearances of x in s. For general x, this is not necessarily the case, because $\mathrm{ord}_x(s)$ counts non-overlapping occurrences only.

For strings $s, x \in \Sigma^*$ we define the *string spectrum* $\mathcal{S}(s,x)$ of s by:

$$\mathcal{S}(s,x) := \big\{y \in \Sigma^* \,:\, \text{there exist } \; a,b \in \Sigma^* \; \text{ with } \; s \in \{yxb, axyxb, axy\}\big\} \cup \{s\} \tag{1}$$

So, the string spectrum $\mathcal{S}(s,x)$ consists of those substrings of s that are bounded by x or by the ends of s. In this context, we call s *sample string* and x *cut string*, while the elements $y \in \mathcal{S}(s,x)$ will be called *fragments* of s (under x).

Example 1. Consider the alphabet $\Sigma := \{\mathbf{0}, \mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}, \mathbf{1}\}$ where the characters $\mathbf{0}$, $\mathbf{1}$ are exclusively used to denote start and end of the sample string. Let $s := \mathbf{0}\mathrm{ACATGTG}\mathbf{1}$ and $x := \mathrm{T}$, then:

$$\mathcal{S}(s, x) = \{\mathbf{0}\mathrm{ACA}, \mathrm{G}, \mathrm{G}\mathbf{1}, \mathbf{0}\mathrm{ACATG}, \mathrm{GTG}\mathbf{1}, \mathbf{0}\mathrm{ACATGTG}\mathbf{1}\}$$

The use of special characters $\mathbf{0}$, $\mathbf{1}$ to uniquely denote start and end of the sample sequence is motivated by the observation that terminal fragments in general differ in mass from inner fragments with otherwise identical sequence, see Lim. (vi). We make use of these characters throughout this paper to reduce the symmetry of the problem, see Example 2 below.

Following (Böcker, 2003), we introduce a mathematical representation of base compositions: We define a *compomer* to be a map $c : \Sigma \to \mathbb{Z}$, where $\mathbb{Z}$ denotes the set of integers. We say that c is a *natural* compomer if $c(\sigma) \geq 0$ holds for all $\sigma \in \Sigma$. For the rest of this paper, we assume that all compomers are natural compomers, unless explicitly stated otherwise. Let $\mathcal{C}_+(\Sigma)$ denote the set of all natural compomers over the alphabet Σ. Clearly, $\mathcal{C}_+(\Sigma)$ is closed with respect to addition, as well as multiplication with a scalar $n \in \mathbb{N}$, where $\mathbb{N}$ denotes the set of natural numbers *including* 0. For finite Σ, in particular, $\mathcal{C}_+(\Sigma)$ is isomorphic to the set $\mathbb{N}^{|\Sigma|}$. We denote the canonical partial order on the set of compomers over Σ by $\preceq$, that is, $c \preceq c'$ if and only if $c(\sigma) \leq c'(\sigma)$ for all $\sigma \in \Sigma$. Furthermore, we denote the *empty compomer* $c \equiv 0$ by 0.

Suppose that $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$, then we use the notation $c = (\sigma_1)_{i_1} \ldots (\sigma_k)_{i_k}$ to represent the compomer $c : \sigma_j \mapsto i_j$ omitting those characters σ_j with $i_j = 0$. For DNA, c represents the number of adenine, cytosine, guanine, and thymine bases in the compomer, and $c = \mathrm{A}_i\mathrm{C}_j\mathrm{G}_k\mathrm{T}_l$ denotes the compomer with $c(\mathrm{A}) = i$, …, $c(\mathrm{T}) = l$. Since the characters $\mathbf{0}$, $\mathbf{1}$ appear at most once in any fragment, we usually omit the indices for these two characters.

The function $\mathrm{comp} : \Sigma^* \to \mathcal{C}_+(\Sigma)$ maps a string $s = s_1 \ldots s_n \in \Sigma^*$ to the compomer of s by counting the number of characters of each type in s:

$$\mathrm{comp}(s) : \Sigma \to \mathbb{N}, \quad \sigma \mapsto \big|\{1 \leq i \leq |s| \,:\, s_i = \sigma\}\big|$$

Note that compomers $\mathrm{comp}(\cdot)$ are also called Parikh-vectors, see (Autebert et al., 1997). The *compomer spectrum* $\mathcal{C}(s, x)$ of s consists of the compomers of all fragments in the string spectrum:

$$\mathcal{C}(s, x) := \mathrm{comp}\big(\mathcal{S}(s, x)\big) = \big\{\mathrm{comp}(y) \,:\, y \in \mathcal{S}(s, x)\big\} \tag{2}$$

For Example 1 we can compute:

$$\mathcal{C}(s, \mathrm{T}) = \{\mathbf{0}\,\mathrm{A}_2\mathrm{C}_1,\ \mathrm{G}_1,\ \mathrm{G}_1\mathbf{1},\ \mathbf{0}\,\mathrm{A}_2\mathrm{C}_1\mathrm{G}_1\mathrm{T}_1,\ \mathrm{G}_2\mathrm{T}_1\mathbf{1},\ \mathbf{0}\,\mathrm{A}_2\mathrm{C}_1\mathrm{G}_2\mathrm{T}_2\mathbf{1}\}$$

Now, the following question arises: For an unknown string s and a known set of cleavage strings $\mathcal{X}$, can we uniquely reconstruct s from its compomer spectra $\mathcal{C}(s, x)$, $x \in \mathcal{X}$? One can easily see that this problem becomes trivial if there exist characters $\mathbf{0}$, $\mathbf{1}$ that uniquely denote the start and end of the sample string — then, for suitable $\mathcal{X}$ like $\mathcal{X} = \Sigma^1 \setminus \{\mathbf{0}, \mathbf{1}\}$, the subsets $\{c \in \mathcal{C}(s, x) \,:\, c(\mathbf{0}) = 1\}$ are sufficient to reconstruct s. This fact was exploited in the Maxam-Gilbert approach (1977). Furthermore, this problem is related to, and appears to be computationally at most as hard as, the well-known Partial Digestion Problem (PDP, see Waterman, 1995): There, one cleaves a sample sequence using restriction enzymes, and measures the lengths of the resulting fragments. It seems likely that we can use algorithms efficiently tackling PDP (Skiena et al., 1990; Skiena and Sundaram, 1994), to solve the above problem in reasonable runtime.

Unfortunately, this approach must fail when applied to experimental MS data, because our theoretical approach of compomer spectra does not take into account the limitations of mass spectrometry and partial cleavage mentioned in the previous section. As we have seen there, Lim. (v) suggests that the probability that some fragment y cannot be detected, strongly depends on the multiplicity of the cut string x as a substring of y. In fact, signals from fragments with $\mathrm{ord}_x(y)$ above a certain threshold will most probably be lost in the noise of

the mass spectrum and, due to Lim. (iii) and (iv), this threshold will be rather small — say, $k \leq 4$ — in real-life applications. This leads us to the following two definitions: For strings s, x and $k \in \mathbb{N} \cup \{\infty\}$, we define the k-string spectrum of s, where k is called the *order* of the string spectrum, by:

$$\mathcal{S}_k(s,x) := \{y \in \mathcal{S}(s,x) \,:\, \mathrm{ord}_x(y) \leq k\} \tag{3}$$

The k-compomer spectrum of s is, in analogy to above, defined by:

$$\mathcal{C}_k(s,x) := \mathrm{comp}\left(\mathcal{S}_k(s,x)\right) = \left\{\mathrm{comp}(y) \,:\, y \in \mathcal{S}(s,x),\ \mathrm{ord}_x(y) \leq k\right\} \tag{4}$$

If the cut string is a single character $x \in \Sigma$, we infer $\mathcal{C}_k(s,x) = \{c \in \mathcal{C}(s,x) \,:\, c(x) \leq k\}$.

For Example 1 we calculate $\mathcal{C}_0(s,\mathrm{T}) = \{\mathbf{0}\,\mathrm{A_2C_1},\ \mathrm{G_1},\ \mathrm{G_1}\mathbf{1}\}$, $\mathcal{C}_1(s,\mathrm{T}) = \mathcal{C}_0(s,\mathrm{T}) \cup \{\mathbf{0}\,\mathrm{A_2C_1G_1T_1},\ \mathrm{G_2T_1}\mathbf{1}\}$, and $\mathcal{C}_2(s,\mathrm{T}) = \mathcal{C}_1(s,\mathrm{T}) \cup \{\mathbf{0}\,\mathrm{A_2C_1G_2T_2}\mathbf{1}\} = \mathcal{C}(s,\mathrm{T})$.

Under what conditions can we uniquely reconstruct a sample string s from its compomer spectra $\mathcal{C}_k(s,x),\ x \in \mathcal{X}$? One can easily see that different strings can share the same k-compomer spectra:

Example 2. Let $\Sigma := \{\mathbf{0}, \mathrm{A}, \mathrm{B}, \mathbf{1}\}$. Then, we cannot uniquely reconstruct the sample string $s = \mathbf{0}\mathrm{BABAAB}\mathbf{1}$ from its complete cleavage compomer spectra $\mathcal{C}_0(s,\mathrm{A})$ and $\mathcal{C}_0(s,\mathrm{B})$, because the string $\mathbf{0}\mathrm{BAABAB}\mathbf{1}$ leads to the same spectra. Analogously, the string $s = \mathbf{0}\mathrm{BABABAABAB}\mathbf{1}$ cannot be reconstructed from its compomer spectra $\mathcal{C}_1(s,x)$ for $x \in \{\mathrm{A}, \mathrm{B}\}$, and we can create such examples for every order k. Furthermore, every string s and its reverse string have identical compomer spectra if all cut strings have length one or — more generally — if all cut strings are symmetric.

Yet, the question stated above does not take into account the problem of false positives: That is, compomers in the set $\mathcal{C}_x$ that do not correspond to actual fragments of the sample sequence. Due to Lim. (ii) and (iv), transforming a mass spectrum into a set of compomers will in general create huge numbers of false positive compomers, since there is usually only one sample fragment corresponding to a peak, but there may be many more compomers with almost identical mass. This number is potentially further increased in view of false positive peaks, see Lim. (iii).

To address this issue we can formulate an optimization problem as follows: For a fixed order $k \in \mathbb{N} \cup \{\infty\}$ let $\mathcal{X} \subseteq \Sigma^*$ be a set of cut strings, and let $\mathcal{C}_x \subseteq \mathcal{C}_+(\Sigma)$ be compomer sets for $x \in \mathcal{X}$. Let $S \subseteq \Sigma^*$ be the set of sample string candidates. Now, find a string $s \in S$ with $\mathcal{C}_k(s,x) \subseteq \mathcal{C}_x$ for all $x \in \mathcal{X}$ that minimizes $\sum_{x\in\mathcal{X}} |\mathcal{C}_x \setminus \mathcal{C}_k(s,x)|$. A potential choice of $S \subseteq \Sigma^*$ are all strings s such that $|s|$ lies in some given interval I, or strings with prefix $\mathbf{0}$ and suffix $\mathbf{1}$. In addition, we may want to search for those strings s of minimal length.

We cannot offer a solution to this problem, but note that the (purely combinatorial) optimization formula $f(s) := \sum_{x\in\mathcal{X}} |\mathcal{C}_x \setminus \mathcal{C}_k(s,x)|$ does not adequately reproduce the experimental "truth": In applications, a tiny peak detected in a mass spectrum can account for several compomers in the corresponding compomer set due to Lim. (iv), and trying to minimize the number of "unused" compomers contradicts the experimental observation. To this end, it makes more sense to find all "good" strings that satisfy at least the inclusion condition:

Sequencing From Compomers (SFC) Problem. For a fixed order $k \in \mathbb{N} \cup \{\infty\}$, let $\mathcal{X} \subseteq \Sigma^*$ be the set of cut strings and, for all $x \in \mathcal{X}$, let $\mathcal{C}_x \subseteq \mathcal{C}_+(\Sigma)$ be a compomer set. Finally, let $S \subseteq \Sigma^*$ be the set of sample string candidates. Now, find all strings $s \in S$ that satisfy $\mathcal{C}_k(s,x) \subseteq \mathcal{C}_x$ for all $x \in \mathcal{X}$.

Note first that this problem differs substantially from PDP and related problems, and approaches for solving PDP cannot be modified to tackle SFC: Such approaches (Skiena et al., 1990; Skiena and Sundaram, 1994) rely on the fact that fragments of s where exactly one base in s has been cleaved, can be detected. Unfortunately, these are precisely the fragments that will always be missing from the compomer sets $\mathcal{C}_x$ due to Lim. (v)! So, SFC is somewhat

"in-between" the Partial Digestion Problem and the Double Digestion Problem (DDP, see Pearson, 1982; Waterman, 1995).

Second, it is trivial to find solutions to the SFC Problem in case $S = \Sigma^*$, because $s = \epsilon$ always satisfies the inclusion conditions. So, S should be chosen to exclude such trivial solutions.

Note that even small compomer sets may lead to a huge number of solutions, that is, exponentially many in the fixed length of the reconstructed strings:

Example 3. Let $\Sigma := \{\mathrm{A}, \mathrm{B}\}$, $k := 0$, and $S := \Sigma^n$ for some $n \in \mathbb{N}$; furthermore $\mathcal{X} := \Sigma^1$, $\mathcal{C}_\mathrm{A} := \{\mathrm{B}_1\}$, and $\mathcal{C}_\mathrm{B} := \{\mathrm{A}_1, \mathrm{A}_2\}$. Every string $s \in S$ that is an arbitrary concatenation $s_0 s_1 s_2 \dots s_k$ where $s_0 \in \{\mathrm{A}, \mathrm{AA}\}$ and $s_j \in \{\mathrm{BA}, \mathrm{BAA}\}$ for $j = 1, \dots, k$ satisfies the conditions $\mathcal{C}_0(s, \mathrm{A}) \subseteq \mathcal{C}_\mathrm{A}$ and $\mathcal{C}_0(s, \mathrm{B}) \subseteq \mathcal{C}_\mathrm{B}$.

In applications, SFC is of interest because there hopefully are fewer solutions for experimental data and, as mentioned above, the optimization problem we introduced, does not capture all aspects of the connection between sample sequences and mass spectra. A reasonable approach here is to judge every solution $s \in S$ of SFC by, say, an adequate probability measure.

3.2. **The undirected sequencing graph.** In this section, we introduce undirected sequencing graphs to tackle the SFC Problem of order $k = 1$. We shall see in the following section that this concept can be seen as a special case of the more elaborate directed sequencing graphs. For the sake of lucidity, we concentrate on the undirected case first.

In the following, we often limit our attention to cut strings x of length 1 to simplify our constructions. Doing so, we still cover all (bio-)chemical cleavage reactions mentioned in Section 2. We indicate in Section 6 how to extend our constructions to arbitrary cut strings $x \in \Sigma^*$. Note that we do not distinguish between the character $x \in \Sigma$ and the corresponding string of length 1.

An (undirected) *graph* consists of a set V of vertices, and a set $E \subseteq \binom{V}{2} \cup V = \big\{\{u, v\} : u, v \in V\big\}$ of edges. An edge $\{v\}$ for $v \in V$ is called a *loop*. We suppose that such graphs are *finite*, that is, have finite vertex set. A *walk* in G is a finite sequence $p = (p_0, p_1, \dots, p_n)$ of elements from V with $\{p_{i-1}, p_i\} \in E$ for all $i = 1, \dots, n$. Note that p is in general not a path because $p_0, \dots, p_n$ do not have to be pairwise distinct. We still use the letter p to denote a walk for convenience. The number $|p| := n$ is defined to be the *length* of p.

Let $\mathcal{C} \subseteq \mathcal{C}_+(\Sigma)$ be an arbitrary set of compomers, and let $x \in \Sigma$ be a single cut string of length one. We define the *undirected sequencing graph* $G_\mathrm{u}(\mathcal{C}, x) = (V, E)$ as follows: The vertex set V consists of all compomers $c \in \mathcal{C}$ such that $c(x) = 0$ holds. The edge set E consists of those $\{u, v\}$ with $u, v \in V$ that satisfy:

$$u + \mathrm{comp}(x) + v \quad \in \mathcal{C} \tag{5}$$

The vertices u, v are not required to be distinct in this equation.

Example 4. For $\Sigma := \{\mathbf{0}, \mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}, \mathbf{1}\}$, $s := \mathbf{0}\mathrm{CTAATCATAGTGCTG}\mathbf{1}$, and $x := \mathrm{T}$ we can calculate the compomer spectrum of order 1:

$$\begin{aligned}\mathcal{C} := \mathcal{C}_1(s, \mathrm{T}) = \{&\mathbf{0}\mathrm{C}_1, \mathbf{0}\mathrm{A}_2\mathrm{C}_1\mathrm{T}_1, \mathrm{A}_2, \mathrm{A}_3\mathrm{C}_1\mathrm{T}_1, \mathrm{A}_1\mathrm{C}_1,\\ &\mathrm{A}_2\mathrm{C}_1\mathrm{G}_1\mathrm{T}_1, \mathrm{A}_1\mathrm{G}_1, \mathrm{A}_1\mathrm{C}_1\mathrm{G}_2\mathrm{T}_1, \mathrm{C}_1\mathrm{G}_1, \mathrm{C}_1\mathrm{G}_2\mathrm{T}_1\mathbf{1}, \mathrm{G}_1\mathbf{1}\}\end{aligned}$$

We have depicted the corresponding sequencing graph $G_\mathrm{u}(\mathcal{C}, \mathrm{T})$ in Figure 2.

How are sequencing graphs related to the SFC Problem? To this end, we say that a string $s \in \Sigma^*$ is 1-*compatible* with a compomer set $\mathcal{C} \subseteq \mathcal{C}_+(\Sigma)$ under $x \in \Sigma^1$ if $\mathcal{C}_1(s, x) \subseteq \mathcal{C}$ holds. Then s is a solution to the SFC Problem of order 1 with respect to x. And, we say that s is *compatible* with a walk $p = p_0 \dots p_l$ in the sequencing graph $G_\mathrm{u}(\mathcal{C}, x)$ if there exist strings $s_0, \dots, s_l \in \Sigma^*$ such that

$$s = s_0 x s_1 x s_2 x \dots x s_l \tag{6}$$

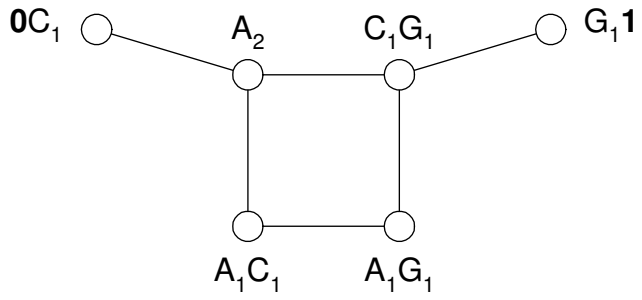

FIGURE 2. The sequencing graph $G_{\mathrm{u}}(\mathcal{C}, \mathrm{T})$ from Example 4.

satisfying $l = |p|$ and $\mathrm{comp}(s_j) = p_j$ for $j = 0, \ldots, l$. This definition implies $\mathrm{ord}_x(s_j) = 0$ for $j = 0, \ldots, l$. We call strings $s_0, \ldots, s_l$ satisfying (6) and $\mathrm{ord}_x(s_j) = 0$ for all $j = 0, \ldots, l$ an *x-partitioning* of s. For $x \in \Sigma^1$, there exists exactly one x-partitioning of s.

In Example 4, the walk ($\mathbf{0}\mathrm{C}_1$, A_2, $\mathrm{A}_1\mathrm{C}_1$, $\mathrm{A}_1\mathrm{G}_1$, $\mathrm{G}_1\mathbf{1}$) is compatible with our input sequence s, but other sequences like $\mathbf{0}$CTAATCGTG$\mathbf{1}$ or $\mathbf{0}$CTAATCGTGATGCTG$\mathbf{1}$ are also compatible with walks in $G_{\mathrm{u}}(\mathcal{C}, \mathrm{T})$.

The next lemma follows from the above definitions, see Lemma 3 for a proof:

Lemma 1. *Let $s \in \Sigma^*$ be a string and $\mathcal{C} \subseteq \mathcal{C}_+(\Sigma)$ a set of compomers. Then, s is 1-compatible with $\mathcal{C}$ under $x \in \Sigma^1$ if and only if there exists a walk p in $G_{\mathrm{u}}(\mathcal{C}, x)$ such that s is compatible with p. Furthermore, this walk p is unique.*

Proposition 2. *For every walk p of a sequencing graph $G_{\mathrm{u}}(\mathcal{C}, x)$ there exist one or more sequences $s \in \Sigma^*$ that are compatible with p and, hence, 1-compatible with $\mathcal{C}$.*

Although basic, the above lemma allows us to search for all strings 1-compatible with a compomer set by simply building all walks in a graph if our set of cut strings $\mathcal{X}$ equals $\Sigma^1 \setminus \{\mathbf{0}, \mathbf{1}\}$: Let $\mathcal{C}_x$ be compomer sets and let p_x be walks in $G_{\mathrm{u}}(\mathcal{C}_x, x)$ for all $x \in \mathcal{X}$. If a sample string $s \in \Sigma^*$ is compatible with p_x for every $x \in \mathcal{X}$, then s is uniquely determined by this property: For the prefix x of s of length 1, we infer from (6) that for every string s' compatible with p_x, x is also a prefix of s'. Repeating this argument leads to $s = s'$ as claimed.

We do not provide an algorithm here to build 1-compatible strings based on the above observation but refer the reader to the next section where we tackle the more general case of directed sequencing graphs.

3.3. **The directed sequencing graph.** A *directed graph* consists of a set V of vertices and a set $E \subseteq V^2 = V \times V$ of edges. An edge (v, v) for $v \in V$ is called a *loop*. Again, we limit our attention to finite directed graphs with finite vertex sets. A *walk* in G is a finite sequence $p = (p_0, p_1, \ldots, p_n)$ of elements from V with $(p_{i-1}, p_i) \in E$ for all $i = 1, \ldots, n$, and $|p| := n$ denotes the *length* of p.

The directed sequencing graphs defined below will be edge-induced subgraphs of the de Bruijn graph (de Bruijn, 1946): For an alphabet Σ and an order $k \geq 1$, the *de Bruijn graph* $B_k(\Sigma)$ is a directed graph with vertex set $V = \Sigma^k$ and edge set

$$E = \{(u, v) \in V^2 \,:\, u_{j+1} = v_j \text{ for all } j = 1, \ldots, k-1\}$$

where $u = (u_1, \ldots, u_k)$ and $v = (v_1, \ldots, v_k)$. In the following, we denote an edge $\big((e_1, \ldots, e_k), (e_2, \ldots, e_{k+1})\big)$ of $B_k(\Sigma)$ by $(e_1, \ldots, e_{k+1})$ for short.

For an arbitrary set of compomers $\mathcal{C} \subseteq \mathcal{C}_+(\Sigma)$ and a cut string $x \in \Sigma$ of length one, we define the *directed sequencing graph* $G_k(\mathcal{C}, x)$ of order $k \geq 1$ as follows: $G_k(\mathcal{C}, x)$ is an edge-induced sub-graph of $B_k(\Sigma_x)$ where

$$\Sigma_x := \{c \in \mathcal{C} \,:\, c(x) = 0\}\,, \tag{7}$$

and an edge $e = (e_1, \ldots, e_{k+1})$ of $B_k(\Sigma_x)$ belongs to $G_k(\mathcal{C}, x)$ if and only if the following condition holds:

$$e_i + c_x + e_{i+1} + c_x + \cdots + c_x + e_{j-1} + c_x + e_j \ \in \mathcal{C} \quad \text{for all } 1 \le i \le j \le k+1 \tag{8}$$

where $c_x := \mathrm{comp}(x)$. By definition, the vertex set of $G_k(\mathcal{C}, x)$ is a subset of $(\Sigma_x)^k$.

Example 5. Let $\mathcal{C} := \mathcal{C}_2(s, \mathrm{T})$, where $s = \mathbf{0}\mathrm{CTAATCATAGTGCTG}\mathbf{1}$ was defined in Example 4. We have depicted the directed sequencing graph $G_2(\mathcal{C}, \mathrm{T})$ in Figure 3. Note that there exist two paths connecting $\mathbf{0}\mathrm{C}_1$ and $\mathrm{G}_1\mathbf{1}$ in $G_{\mathrm{u}}(\mathcal{C}, \mathrm{T})$, but only one directed walk from $(\mathbf{0}\mathrm{C}_1, \mathrm{A}_2)$ to $(\mathrm{C}_1\mathrm{G}_1, \mathrm{G}_1\mathbf{1})$ in $G_2(\mathcal{C}, \mathrm{T})$: The ambiguity of the compomer $\mathrm{A}_2\mathrm{C}_1\mathrm{G}_1\mathrm{T}_1$ is resolved in $\mathcal{C}_2(s, \mathrm{T})$ by the existence of compomers $\mathrm{A}_4\mathrm{C}_1\mathrm{G}_1\mathrm{T}_2$ and $\mathrm{A}_2\mathrm{C}_2\mathrm{G}_2\mathrm{T}_2$, and the non-existence of compomers $\mathbf{0}\mathrm{A}_2\mathrm{C}_2\mathrm{G}_1\mathrm{T}_2$ and $\mathrm{A}_2\mathrm{C}_1\mathrm{G}_2\mathrm{T}_2\mathbf{1}$.

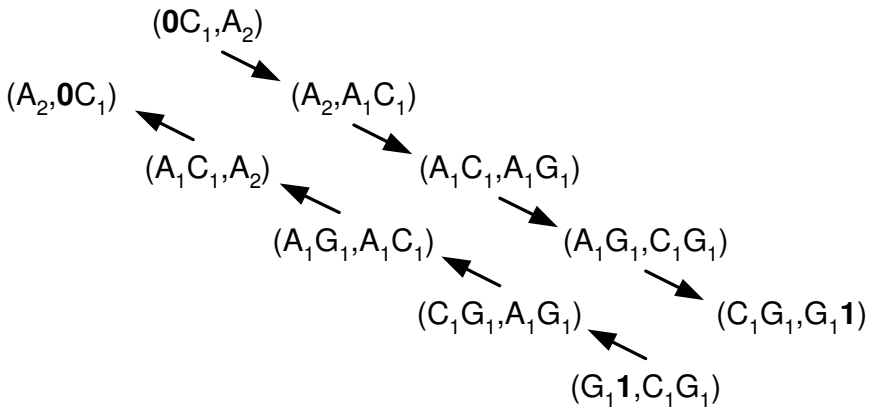

FIGURE 3. The directed sequencing graph $G_2(\mathcal{C}, \mathrm{T})$ from Example 5.

We want to point out that the alphabet Σ_x of the underlying de Bruijn graph does not coincide with the sequence alphabet Σ, unlike the Sequencing By Hybridization approach introduced by Pevzner (1989), but instead consists of those compomers $c \in \mathcal{C}$ with $c(x) = 0$. The main distinction between SFC and SBH, though, is that we search the de Bruijn graph for walks instead of Eulerian paths:

- We have to deal with many false positive edges here, because of "noise" peaks, misinterpreted peaks, and misinterpreted compomers.
- We do not know the multiplicity of compomers in $\mathcal{C}_x$, and compomers $c \in \mathcal{C}_x$ of small "order" will regularly correspond to two or more fragments of the sample sequence.

Analogously to the previous section, we say that a string $s \in \Sigma^*$ is *k-compatible* with a compomer set $\mathcal{C} \subseteq \mathcal{C}_+(\Sigma)$ under $x \in \Sigma^1$ if $\mathcal{C}_k(s, x) \subseteq \mathcal{C}$ holds. Note that by definition, such a string s satisfies the condition of the Sequencing From Compomers Problem of order k. The string s is called *compatible* with a walk $p = p_0 \ldots p_{|p|}$ in the sequencing graph $G_k(\mathcal{C}, x)$ if the x-partitioning $s_0, \ldots, s_l \in \Sigma^*$ of s from (6) satisfies $l = |p| + k - 1$ and

$$p_j = \big(c_j, c_{j+1}, \ldots, c_{j+k-1}\big) \quad \text{for } j = 0, \ldots, |p| \ , \tag{9}$$

where $c_j := \mathrm{comp}(s_j)$ for $j = 0, \ldots, l$. Recall that $\mathrm{ord}_x(s_j) = 0$ must hold for $j = 0, \ldots, |p|$. We note that the definitions of "1-compatible" in the previous section, and "k-compatible" for $k = 1$ are equivalent, and so are the graphs $G_{\mathrm{u}}(\mathcal{C}, x)$ and $G_1(\mathcal{C}, x)$: For every edge $\{u, v\}$ with $u \neq v$ in $G_{\mathrm{u}}(\mathcal{C}, x)$ there exist two edges (u, v) and (v, u) in $G_1(\mathcal{C}, x)$, and for every loop $\{v\}$ in $G_{\mathrm{u}}(\mathcal{C}, x)$ there exists a loop (v, v) in $G_1(\mathcal{C}, x)$.

Example 6. For $s := \mathbf{0}\mathrm{BABABABAABABAB}\mathbf{1}$ created analogously to Example 2, the graph $G_2(\mathcal{C}, \mathrm{B})$ for $\mathcal{C} := \mathcal{C}_2(s, \mathrm{B})$ is depicted in Figure 4. If we remove the (superfluous) vertices

$(\mathrm{A}_1, \mathbf{0})$ and $(\mathbf{1}, \mathrm{A}_1)$, there still exist two walks of length 6 from $(\mathbf{0}\mathrm{A}_1, \mathrm{A}_1)$ to $(\mathrm{A}_1, \mathrm{A}_1\mathbf{1})$ that traverse all edges of the resulting graph; the two sequences compatible with these two walks are our initial sequence, plus the "stutter" sequence **0**BABABAABABABAB**1**, respectively: The positions 7 and 8 of s are exchanged in this string.

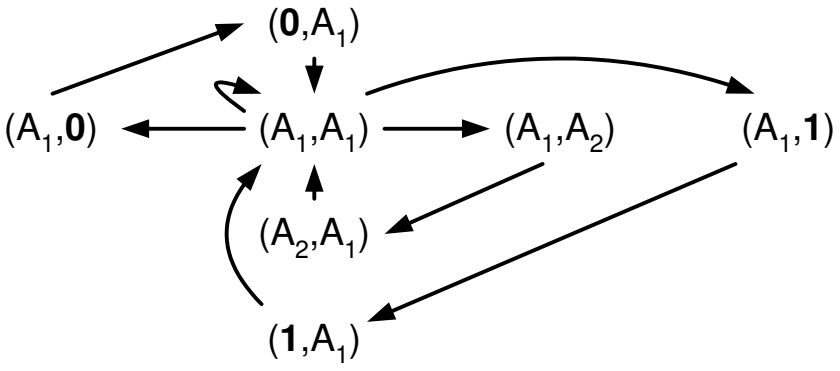

FIGURE 4. The directed sequencing graph $G_2(\mathcal{C}, \mathrm{B})$ from Example 6.

Lemma 3. *Let $s \in \Sigma^*$ be a string with $\mathrm{ord}_x(s) \geq k$ for $x \in \Sigma^1$, and let $\mathcal{C} \subseteq \mathcal{C}_+(\Sigma)$ be a set of compomers. Then, s is k-compatible with $\mathcal{C}$ under x if and only if there exists a walk p in the sequencing graph $G_k(\mathcal{C}, x)$ such that s is compatible with p. Furthermore, this walk p is unique.*

Proposition 4. *For every walk p of a sequencing graph $G_k(\mathcal{C}, x)$ there exist one or more sequences $s \in \Sigma^*$ compatible with p and, hence, k-compatible with $\mathcal{C}$.*

It is straightforward to derive the proof of Lemma 3 from the definitions:

Proof of Lemma 3. Suppose that s is k-compatible with $\mathcal{C}$, and let $s_0, \ldots, s_l \in \Sigma^*$ be the x-partitioning of s. Using equation (9) we can define $l-k+1$ compomers $p_0, \ldots, p_{l-k+1} \in \mathcal{C}^k$, then the definition of $\mathcal{C}_k(s, x)$ implies $\{\mathrm{comp}(s_j) : j = 0, \ldots, l\} \subseteq \Sigma_x$ and, hence, that $p = (p_0, \ldots, p_{l-k+1})$ is a walk in the de Bruijn graph $B_k(\Sigma_x)$. By definition of $\mathcal{C}_k(s, x)$, (8) must hold for every edge $(e_1, \ldots, e_{k+1})$ of p, so p is also a walk in $G_k(\mathcal{C}, x)$ as claimed. If s is compatible with paths p, p' in $G_k(\mathcal{C}, x)$ then (9) implies $p = p'$.

Suppose now that s is compatible with a walk p in $G_k(\mathcal{C}, x)$. Let $s_0, \ldots, s_l \in \Sigma^*$ be the x-partitioning of s, then $l = |p| - k + 1$ must hold. Let $y \in \mathcal{S}(s, x)$ be a fragment of s with $\mathrm{ord}_x(y) \leq k$, then there exist indices i_0, j_0 with $j_0 - i_0 \leq k$ such that $y = s_{i_0} x \ldots x s_{j_0}$. Let $j := \min\{i_0, |p| - k\}$ then by (9), the $(j+1)$-th edge of p is

$$e = \Big(\big(\mathrm{comp}(s_j), \ldots, \mathrm{comp}(s_{j+k-1})\big), \big(\mathrm{comp}(s_{j+1}), \ldots, \mathrm{comp}(s_{j+k})\big) \Big)$$

and, since e is an edge of $G_k(\mathcal{C}, x)$, we infer from (8) that

$$\mathrm{comp}(s_{i_0}) + \mathrm{comp}(x) + \cdots + \mathrm{comp}(x) + \mathrm{comp}(s_{j_0}) \in \mathcal{C}$$

must hold and, hence, $\mathrm{comp}(y) \in \mathcal{C}$. We conclude $\mathcal{C}_k(s, x) \subseteq \mathcal{C}$ as claimed. □

For a given set of compomers, how sparse is the corresponding sequencing graph in general? Clearly, the de Bruijn graph of order k over the alphabet Σ_x has $|\Sigma_x|^k$ vertices and $|\Sigma_x|^{k+1}$ edges. Unfortunately, the number of vertices and edges of a sequencing graph may be of the same order as those of the de Bruijn graph itself even for small compomer sets of size $O(k\,|\Sigma_x|)$ and "short" strings of length $O(n^2 + kn)$, as the following two lemmata show:

Lemma 5. *Let Σ be an alphabet of size $|\Sigma| \geq 2$, $x \in \Sigma^1$ a cut string of length one, and let $k \in \mathbb{N}$ be the fixed order. Then, for every $n \in \mathbb{N}$ there exist compomer sets $\mathcal{C}_n \subseteq \mathcal{C}_+(\Sigma)$ satisfying*

$$|\mathcal{C}_n| = (k+1)(n+1) \quad \text{and} \quad |\Sigma_x| = n+1 \text{ for } \Sigma_x = \{c \in \mathcal{C}_n \,:\, c(x) = 0\} \tag{10}$$

such that the corresponding sequencing graph $G_k(\mathcal{C}_n, x)$ has $\binom{n+k}{k} = \Theta(n^k)$ vertices and $\binom{n+k+1}{k+1} = \Theta(n^{k+1})$ edges.

We omit the proofs of this and the following lemma and only note that the sets

$$\mathcal{C}_n := \{\mathrm{A}_m\mathrm{B}_i \,:\, m = 0, \ldots, n \text{ and } i = 0, \ldots, k\} \tag{11}$$

satisfy the conditions of the above lemma for $x := \mathrm{B}$.

Lemma 6. *There exist strings $s \in \Sigma^*$ over the alphabet $\Sigma = \{\mathrm{A}, \mathrm{B}\}$ of length $O(n^2 + kn)$ with $\mathcal{C}_k(s, \mathrm{B}) = \mathcal{C}_n$ as defined in* (11).

In view of Lemmata 5 and 6, one can suspect that it is impossible to perform de-novo sequencing from compomers. Fortunately, this seems not to be the case neither for random sequences nor for biological sequence data, see Section 5.

4. Algorithm

Let Σ be a constant and finite alphabet where $\mathbf{0}, \mathbf{1} \in \Sigma$ uniquely denote the first and last character of our sample strings. Let $\mathcal{X} = \Sigma^1 \setminus \{\mathbf{0}, \mathbf{1}\}$ be the set of cut strings, and $k \in \mathbb{N}$ the fixed order. We are given sets of compomers $\mathcal{C}_x$ for $x \in \mathcal{X}$ and a set $S \subseteq \Sigma^*$ of strings, and want to solve the Sequencing From Compomers Problem, that is, find all sample strings $s \in S$ satisfying $\mathcal{C}_k(s, x) \subseteq \mathcal{C}_x$ for all $x \in \mathcal{X}$. We further concentrate on the case especially relevant for applications, where

$$S = \left\{ s \in \Sigma^* \,:\, l_{\min} \leq |s| \leq l_{\max}, \text{ and } s = \mathbf{0}\, s'\, \mathbf{1} \text{ for some } s' \in (\Sigma \setminus \{\mathbf{0}, \mathbf{1}\})^* \right\} \tag{12}$$

contains all strings of length in a given interval. This is because we either know the approximate length of the unknown string due to our experimental setup, or we can easily estimate it if necessary.

To solve SFC, we present a depth-first search that backtracks through sequence space, moving along the edges of the sequencing graphs in parallel. In this way, we implicitly build walks in the directed sequencing graphs of order k that are compatible with the constructed sequences. Because of Lemma 3, these sequences are in fact k-compatible with $\mathcal{C}_x$ under x for every $x \in \mathcal{X}$ and, hence, are solutions to SFC. In every recursion step of the algorithm, we attach every character $x \in \mathcal{X}$ to the previously known string s. This forces us to do an edge transition in the sequencing graph $G_k(\mathcal{C}_x, x)$, and we can stop the recursion if this edge transition is not possible. In addition, we do another branch-and-bound check by testing if it will be possible in the future to do edge transitions in all other sequencing graphs.

4.1. **Building the sequencing graphs.** First, we have to build the sequencing graphs $G_x := G_k(\mathcal{C}_x, x)$ for $x \in \mathcal{X}$. For $\Sigma_x := \{c \in \mathcal{C}_x \,:\, c(x) = 0\}$ we search for all those vectors $e \in (\Sigma_x)^{k+1}$ that satisfy (8). We make use of the trivial approach here: For every $(k+1)$-tuple $(e_1, \ldots, e_{k+1}) \in (\Sigma_x)^{k+1}$ we test if it satisfies equation (8) with $\mathcal{C} = \mathcal{C}_x$. This can be performed in $O\big(|\Sigma_x|^{k+1}\, k^2\big)$ time using a hash table to check $c \in \mathcal{C}_x$ and, since Σ_x and k are small in applications, this approach is sufficient here. If the condition is satisfied, we add $(e_1, \ldots, e_k)$ and $(e_2, \ldots, e_{k+1})$ to the vertex set of G_x, and we add $(e_1, \ldots, e_{k+1})$ to the edge set of G_x. A faster algorithm for building $G_k(\mathcal{C}_x, x)$ is to iteratively build the graphs $G_\kappa(\mathcal{C}_x, x)$ for $\kappa = 1, \ldots, k$.

For applications, we have to slightly modify the construction of our sequencing graphs G_x: Firstly, $G_k(\mathcal{C}_x, x)$ contains superfluous edges and vertices (cf. Example 6), because we know from (12) that characters $\mathbf{0}, \mathbf{1}$ are uniquely used to denote beginning and end of any string

$s \in S$. So, we can limit the above calculations to edges $e = (e_1, \dots, e_{k+1})$ such that $e_1(\mathbf{0}) \leq 1$, $e_{j+1}(\mathbf{0}) = e_j(\mathbf{1}) = 0$ for $j = 1, \dots, k$, and $e_{k+1}(\mathbf{1}) \leq 1$.

More important, there remains one last problem: We do not know where to start in the sequencing graph! To this end, let $* \notin \Sigma_x$ denote a special source character. We add the source vertex $(*, \dots, *)$ to G_x, but further edges and vertices are necessary to enter the "regular" part of the graph: A *source edge* is an edge $e = (e_1, \dots, e_{k+1})$ of the de Bruijn graph $B_k\big(\Sigma_x \cup \{*\}\big)$ such that there exists some $\kappa \in \{2, \dots, k+1\}$ with:

- $e_j = *$ for $j = 1, \dots, \kappa - 1$, and $e_j \neq *$ for $j = \kappa, \kappa + 1, \dots, k + 1$
- $e_\kappa(\mathbf{0}) = 1$, $e_{j+1}(\mathbf{0}) = e_j(\mathbf{1}) = 0$ for $j = \kappa, \kappa + 1, \dots, k$, and $e_{k+1}(\mathbf{1}) \leq 1$
- Equation (8) holds *only for* $\kappa \leq i \leq j \leq k + 1$

We add all source edges to the edge set of G_x, and we add all induced vertices to the vertex set of G_x: These induced vertices are of the form $(*, \dots, *, v_\kappa, \dots, v_k)$ where $2 \leq \kappa \leq k$ and $v_\kappa(\mathbf{0}) = 1$. Note that the resulting graph G_x is a subgraph of $B_k\big(\Sigma_x \cup \{*\}\big)$. Now, we can use the source vertex $(*, \dots, *)$ of G_x as our start vertex v_x^{start}.

We do not have to explicitly construct a sink, since the recursion below terminates as soon as we can add the end character $\mathbf{1}$. Note again that due to Lim. (vi), the masses of fragments from beginning and end of the sample string in general differ from those of all other fragments in applications, what has to be taken into account when computing the sets $\mathcal{C}_x$.

4.2. **The depth-first search.** Now, we start the recursion with the string $s := \mathbf{0}$. We initialize the current vertices $v_x := v_x^{\text{start}}$ for all $x \in \mathcal{X}$.

In the recursion step, let s be the current sample string, and for all $x \in \mathcal{X}$, let v_x be the current active vertices in the sequencing graph G_x. Let s_x be the unique string satisfying $\operatorname{ord}_x(s_x) = 0$ such that either xs_x is a suffix of s, or $s_x = s$ if $\operatorname{ord}_x(s) = 0$. Set $c_x := \operatorname{comp}(s_x)$.

- If $|s| + 1 \geq l_{\min}$ and we can do an edge transition to an end vertex in all sequencing graphs G_x for $x \in \mathcal{X}$, then **output** $s\mathbf{1}$ as a sequence candidate.
- If $|s| < l_{\max}$, then let $\Sigma_{\text{a}} \subseteq \Sigma$ be the set of admissible characters as defined below. For every admissible character $x \in \Sigma_{\text{a}}$ do a recursion step: Replace s by the concatenation sx; and in the sequencing graph G_x, replace the active vertex $v_x = (v_1, v_2, \dots, v_k)$ by $(v_2, \dots, v_k, c_x)$ that is a vertex of G_x.
- Return to the previous level of recursion.

Here we call a character $x \in \mathcal{X}$ *admissible* if the following two conditions hold:

- Let $v_x = (v_1, \dots, v_k)$ be the active vertex in G_x. Then, the $(k+1)$-tuple $(v_1, \dots, v_k, c_x)$ must be an edge of the sequencing graph G_x.
- For every $\sigma \in \mathcal{X} \setminus \{x\}$, let $v_\sigma = (v_1, \dots, v_k)$ be the active vertex in G_σ. Then, there must exist at least one edge $(v_1, \dots, v_k, c'_\sigma)$ in the sequencing graph G_σ such that $c_\sigma \preceq c'_\sigma$ holds.

We say that we can perform an *edge transition to an end vertex* in a sequencing graph G_x for $x \in \mathcal{X}$ if the following holds: Let $v_x = (v_1, \dots, v_k)$ be the active vertex in G_x. Set $c'_x := c_x + \mathbf{1}_1$, where $\mathbf{1}_1$ denotes the compomer containing exactly one end character. Then, the $(k+1)$-tuple $(v_1, \dots, v_k, c'_x)$ must be an edge of the sequencing graph G_x.

Theorem 1. *For fixed order k, $\mathcal{X} := \Sigma \setminus \{\mathbf{0}, \mathbf{1}\}$, and S as defined in* (12)*, the algorithm of this section solves the Sequencing From Compomers Problem by returning all strings $s \in S$ that satisfy $\mathcal{C}_k(s, x) \subseteq \mathcal{C}_x$ for all $x \in \mathcal{X}$.*

Proof. It is clear from the construction that any output string s is compatible with a walk in G_x and, by Lemma 3, k-compatible with $\mathcal{C}_x$ for all $x \in \mathcal{X}$.

It remains to be shown that all strings $s \in S$ that are k-compatible with $\mathcal{C}_x$ for all $x \in \mathcal{X}$, are constructed by the algorithm. To this end, let $s \in S$ be such a string. By Lemma 3, there exist unique walks $p^x = p_0^x\, p_1^x \dots p_r^x$ in G_x compatible with s, for every $x \in \mathcal{X}$. We will show by induction that every proper prefix s' of s is an input to the recursion step of the algorithm.

This implies that s' with $s'\mathbf{1} = s$ is also an input of the recursion step. Analogously to the reasoning below, one can show that at this point, we can perform edge transitions to an end vertex in every sequencing graph. It follows that $s'\mathbf{1} = s$ is an output of the algorithm.

The induction basis is trivial for $s' = \mathbf{0}$. Assume that $s' = \tilde{s}x$ for some $\tilde{s} \in \Sigma^*$ and $x \in \Sigma$. Let $s_0, \ldots, s_l$ be the x-partitioning of $\tilde{s}$ as defined in (6). The uniqueness of the x-partitioning of s implies $\mathrm{comp}(s_j) = p_j^x$ for $j = 0, \ldots, l$. In addition to the claim above, we claim that in the current recursion step, the active vertex in G_x is set to $(p_{l-k+1}^x, \ldots, p_l^x)$. To simplify matters, we ignore the case $l < k$ that one can solve analogously. By the induction hypothesis, we know that $\tilde{s}$ is an input of the recursion step, and that the active vertex in G_x at this point is still $(p_{l-k}^x, \ldots, p_{l-1}^x)$.

We claim that x is admissible: We know that $(p_{l-k}^x, \ldots, p_l^x)$ is an edge of p^x and, hence, of G_x, so the first condition is satisfied. For $\sigma \in X \setminus \{x\}$, note that the active vertex has not changed since the last time we appended σ. Analogously to above, we can show that $p_0^\sigma, \ldots, p_{m-1}^\sigma, c_\sigma$ are the compomers of the σ-partitioning of s'; note again that $\mathrm{ord}_\sigma(s_\sigma) = c_\sigma(\sigma) = 0$. Hence, the active vertex in G_σ is $(p_{m-k}^\sigma, \ldots, p_{m-1}^\sigma)$, and $(p_{m-k}^\sigma, \ldots, p_m^\sigma)$ is an edge of G_σ. Since s' is a prefix of s, we infer $c_\sigma = \mathrm{comp}(s_\sigma) \preceq p_m^\sigma$, so the second condition is satisfied, too. This implies that x is admissible.

It follows directly from the construction of the algorithm that the new active vertex in G_x is set to $(p_{l-k+1}^x, \ldots, p_l^x)$. Consequently, $s' = \tilde{s}x$ is an input of the recursion step, where the new active vertex in G_x is $(p_{l-k+1}^x, \ldots, p_l^x)$, as claimed. □

What are time and space requirements of the described algorithm? Clearly, there can be exponentially many solutions to SFC (cf. Example 3), so the worst-case runtime is also exponential in the problem size as well as the maximal length of an output string. In addition, the runtime can still be exponential if there is a unique solution to SFC, or no solution at all. On the contrary, the space requirements are rather moderate: We need $O(m^{k+1})$ memory to store the sequencing graphs, where $m := \max\{|\Sigma_x| : x \in \mathcal{X}\}$. For $n := \max\{|s| : s \in S\}$ we need $O(n)$ memory in the recursion part of the algorithm, because every recursion step uses only constant memory: The reconstructed sequence itself is not stored in the recursion step but only the current character. The critical factor is obviously storing the sequencing graphs, but note that in applications, these graphs are supposedly sparse and a suitable graph representation will allow storing such graphs with much less memory requirements than the worst-case $O(m^{k+1})$ suggests.

The complete process of de-novo sequencing from mass spectrometry data can now be performed as follows: For every cleavage reaction, apply a peak detection algorithm to extract those parts of the measured mass spectrum that most probably correlate to particles in our sample. For every detected peak, calculate all compomers with at most k cleavage bases and with mass sufficiently close to that of the detected peak. Note that we can limit these calculations to compomers containing at most one character $\mathbf{0}, \mathbf{1}$. In this way, we generate compomer sets $\mathcal{C}_x$ for all $x \in \mathcal{X}$. As described in Section 4.1, we build sequencing graphs G_x from these compomer sets. We use the algorithm of Section 4.2 to generate all sequence candidates that are k-compatible with our input compomer sets $\mathcal{C}_x$.

After generating all sequence candidates in this fashion, we want to further evaluate these sequence candidates taking into account the mass spectrometry data available from all cleavage reactions. A simple scoring scheme for doing so was described in (Böcker, 2003), where only slight modifications are needed to deal with partial cleavage data. A more advanced scoring scheme could compute likelihood values for the model [reference sequence is s] to calculate a score for every sequence candidate s. Such scoring schemes can use additional information such as peak intensities, overlapping peaks, and peaks corresponding to fragments of order $> k$ to discriminate between sequence candidates. We do not go into the details of this problem here.

Note that for the application of DNA sequencing, we cannot circumvent the complexity of SFC by, say, introducing heuristics with good time complexity: Such heuristics might or might not find (all of) the solutions of SFC. But this is not acceptable in the setting of de-novo sequencing.

5. Results

In the following, we report some preliminary results of our approach; a more detailed evaluation is currently in progress.

In the absence of sufficient data to test the algorithm, we simulated cleavage reactions and mass spectra, and examined the performance of the algorithm from the previous section on this simulated data. We used two data sets to generate the sample DNA: First, we generated random sample DNA sequences proposing that all bases have identical frequency $\frac{1}{4}$ of occurrence. Second, we used the Human LAMB1 gene (`ENSG00000091136`, see Reich et al., 2001) and chopped it into approximately 400 pieces, using both exons and introns. For a preliminary evaluation of the method, we performed simulations for sequence length $l = 200$ and order $k = 2$, only. Our initial simulations indicate that the presented approach can be used to tackle SFC.

We simulated four cleavage reactions based on real world RNAse cleavage, where we generated only fragments of order at most $k = 2$, supposing that peaks from fragments of order $k + 1$ and higher cannot be detected in the mass spectrum. Then, we calculated masses of all resulting fragments, and addressed Lim. (ii) (calibration and resolution of the mass spectrometer) in the following way: We say that $\delta \geq 0$ is the *accuracy* of the mass spectrometer, where δ is the maximal difference between an expected and the corresponding detected mass. For our initial evaluation we used $\delta = 0.3$ Da corresponding to OTOF mass spectrometry. We perturbed every signal from the expected list of peaks so that its mass differs by at most δ from the expected mass, and for every resulting peak we calculated all compomers of order at most k that might possibly create a peak with mass at most δ off the perturbed signal mass. In this way, we created the sets $\mathcal{C}_x$ for $x \in \Sigma$. Note again that we do not take into account the intensities of peaks.

When simulating the mass spectrometry analysis, we simulate neither false positives (additional peaks) nor false negatives (missing peaks) here. The former does not change the results dramatically: Every signal peak can potentially be interpreted as many different compomers due to Lim. (iv). Hence, the compomer lists *do* contain many false positives. The latter, on the contrary, makes it necessary to modify our approach to deal with real world data.

We want to reconstruct our sample DNA from the cleavage reaction data using sequencing graphs of order 2 and the algorithm presented in the previous section. We assumed that the length of the sample sequence is known with a relative error of 10%, so we set $l_{\min} := 180$ and $l_{\max} := 220$. In addition, we assumed that 8 bases at the start and end of the sequence were known in advance, so our sample sequences had a total length of 216 nt. As we learned from these simulations, the most common sequencing error of our approach seems to be the exchange of two bases belonging to a "stutter" repeat (cf. Examples 2 and 6).

We present the results of our simulation in Table 1. Here we provide the number of *ambiguous* bases for the given setup: Formally, an ambiguous base is a column in the multiple alignment, taken over all output sequence candidates, where the aligned output sequences differ. One can see that even when the input sequence was not reconstructed uniquely, there are often only a few ambiguities in the output sequences: The average ratio of ambiguous bases was $\frac{0.4}{1000}$ for the random sequences, and $\frac{2.5}{1000}$ for the LAMB1 sequences. As one could have expected, there were no sample sequences with exactly one ambiguous base. By design, the correct sequence was always among the output sequence candidates.

# ambiguous bases	random seq.		LAMB1 seq.	
0	961	(96.4 %)	341	(90.0 %)
2	30	(3.0 %)	22	(5.8 %)
3	1	(0.1 %)	0	(0 %)
4	5	(0.5 %)	4	(1.1 %)
5	0	(0 %)	1	(0.3 %)
6	0	(0 %)	2	(0.5 %)
8	0	(0 %)	2	(0.5 %)
10+	0	(0 %)	6	(1.8 %)
total	997	(100.0 %)	378	(100.0 %)

TABLE 1. Results of the simulations for $k = 2$, $l = 200$, and $\delta = 0.3$. For a number m of ambiguous bases, we have listed the absolute and relative number of input sequences where a unique reconstruction of the sequence was possible ($m = 0$) or not possible ($m > 0$).

6. Discussion and improvements

We have introduced the Sequencing From Compomers Problem that stems from the analysis of mass spectrometry data from partial cleavage experiments. Although this problem is computationally difficult in general, we have introduced a computational approach to perform de-novo sequencing from such data. The introduced method uses sub-graphs of the de Bruijn graph to construct all sequences that are compatible with the observed mass spectra. We tested the performance of our approach on simulated mass spectrometry data from random as well as from biological sequences. Surprisingly, our approach is capable of reconstructing the correct sequence in most cases, and ambiguities are limited most of the time to the exchange of two bases in a "stutter repeat." The local information of compomers derived from substrings of the input sequence that contain only a few (here, $k = 2$) cleavage bases, is often sufficient to reconstruct the input string. Our simulations indicate that the presented approach may enable de-novo sequencing with an experimental setup that differs completely from Sanger Sequencing, and still allows for sequencing lengths that are of the same magnitude as those of Sanger Sequencing. Using base-specific cleavage and mass spectrometry for de-novo sequencing has the advantages of high throughput (4 mass spectra can be measured in less than 10 seconds) and potentially increased sensitivity and specificity over classical Sanger Sequencing. Potential additional advantages are the possibility to perform pooling or multiplexing, see below.

As noted in Section 5, the simulation results of that section are only a first step in evaluating the power of our approach. A more thorough simulation analysis is currently in progress.

We mentioned earlier that our setup can easily be extended to cut strings x of arbitrary length. Moreover, it makes sense to replace the cut bases $x \in \Sigma$ by *sets* of cut strings $X \subseteq \Sigma^*$ because there exist enzymes that are specific to, say, both pyrimidines C or T at a certain position. For example, the enzyme *Hin*fI cleaves all sequences of the form GANTC, and here we define $X := \{\mathrm{GAATC}, \mathrm{GACTC}, \mathrm{GAGTC}, \mathrm{GATTC}\}$. The generalization of the presented tools is pretty much straightforward, but note that (a) we can no longer simply calculate the order of a compomer, and (b) we have to cope with cleavage that happens *within* a cut string.

Our approach does at no point depend on the fact that we are sequencing *DNA*. In theory, another application can be de-novo sequencing of proteins, given that we have ways of amino acid (or cut string) specific cleavage of such polypeptides. In reality, this may be difficult because there are 20 characters in the protein alphabet, so a straightforward generalization would call for 20 distinct cleavage reactions. In addition, calculating all compomers for a given mass becomes computationally difficult. However, protein "sequencing" using tryptic digest and a lookup database is a broadly used tool in Proteomics. Note that our approach differs

fundamentally from MS/MS peptide sequencing approaches (Dančík et al., 1999; Patterson and Aebersold, 1995). Furthermore, our approach does not rely on data from MALDI-TOF MS but will work with any method that allows us to determine base compositions of the cleavage reaction products.

Finally, we want to point out that the presented method can easily be adopted for *pooling* as well as *multiplexing*: When pooling sequences, we want to analyze *mixtures* of samples, or heterozygous samples. Our approach will in principal return all sequences found in the mixture, and only a subsequent analysis has to be modified accordingly. For multiplexing, instead of sequencing a single continuous stretch of the sample sequence of length 200 nt, we analyze, say, ten distinct stretches of length 20 nt each in parallel. Here we can iteratively search for 10 appropriate walks in every sequencing graph.

The intensity of a peak in a MS spectrum may indicate the multiplicity of the respective compomer. This motivates the question whether we can uniquely reconstruct the string s from $\mathcal{C}_k(s, x)$ if we define all sets in equations (1–4) to be multisets instead of simple sets.

Clearly, there are more elaborate ways to use sequencing graphs for solving SFC than the depth-first search algorithm we used. In particular, a depth-first search is not fully appropriate: As we have noted in Section 5, the most common ambiguities are stutter repeats where resulting sequences are compatible with walks that are identical except for a short de-tour element.

On the theoretical side of the problem, it would be interesting to characterize transformations of strings that, like stutter repeats, do not change the compomer spectra of the string for some given order k.

Finally, we have mentioned in the previous section that in this paper, we do not take into account the problem of false negative peaks that is common in applications. This will be addressed in a forthcoming paper.

Acknowledgments

I want to thank Zsuzsanna Lipták, Jens Stoye, Matthias Steinrücken, and Dirk van den Boom for proofreading earlier versions of this manuscript, the latter also for many helpful suggestions.

References

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

Bains, W. and Smith, G. C. (1988). A novel method for nucleic acid sequence determination. *J. Theor. Biol.*, 135:303–307.

Böcker, S. (2003). SNP and mutation discovery using base-specific cleavage and MALDI-TOF mass spectrometry. *Bioinformatics*, 19:i44–i53.

Cosnard, M., Duprat, J., and Ferreira, A. G. (1990). The complexity of searching in $X + Y$ and other multisets. *Information Processing Letters*, 34:103–109.

Dančík, V., Addona, T. A., Clauser, K. R., Vath, J. E., and Pevzner, P. A. (1999). De novo peptide sequencing via tandem mass spectrometry. *J. Comp. Biol.*, 6(3/4):327–342.

de Bruijn, N. G. (1946). A combinatorial problem. In *Indagationes Mathematicae*, volume VIII, pages 461–467. Koninklije Nederlandsche Akademie van Wetenschappen.

Drmanac, R., Labat, I., Brukner, I., and Crkvenjakov, R. (1989). Sequencing a megabase plus DNA by hybridization: Theory of the method. *Genomics*, 4:114–128.

Hartmer, R., Storm, N., Böcker, S., Rodi, C. P., Hillenkamp, F., Jurinke, C., and van den Boom, D. (2003). RNAse T1 mediated base-specific cleavage and MALDI-TOF MS for high-throughput comparative sequence analysis. *Nucl. Acids. Res.*, 31(9):e47.

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

Köster, H., Tang, K., Fu, D.-J., Braun, A., van den Boom, D., Smith, C. L., Cotter, R. J., and Cantor, C. R. (1996). A strategy for rapid and efficient DNA sequencing by mass spectrometry. *Nat. Biotechnol.*, 14(9):1084–1087.

Lysov, Y., Floretiev, V., Khorlyn, A., Khrapko, K., Shick, V., and Mirzabekov, A. (1988). DNA sequencing by hybridization with oligonucleotides. *Dokl. Acad. Sci. USSR*, 303:1508–1511.

Maxam, A. M. and Gilbert, W. (1977). A new method for sequencing DNA. *Proc. Nat. Acad. Sci. USA*, 74(2):560–564.

Patterson, S. D. and Aebersold, R. (1995). Mass spectrometric approaches for the identification of gel-separated proteins. *Electrophoresis*, 16:1791–1814.

Pearson, W. R. (1982). Automatic construction of restriction site maps. *Nucleic Acids Res.*, 10:217–227.

Pevzner, P. P. (1989). l-tuple DNA sequencing: Computer analysis. *J. Biomol. Struct. Dyn.*, 7:63–73.

Reich, D. E., Cargill, M., Bolk, S., Ireland, J., Sabeti, P. C., Richter, D. J., Lavery, T., Kouyoumjian, R., Farhadian, S. F., Ward, R., and Lander, E. S. (2001). Linkage disequilibrium in the human genome. *Nature*, 411:199–204.

Rodi, C. P., Darnhofer-Patel, B., Stanssens, P., Zabeau, M., and van den Boom, D. (2002). A strategy for the rapid discovery of disease markers using the MassARRAY system. *BioTechniques*, 32:S62–S69.

Ronaghi, M., Uhlén, M., and Nyrén, P. (1998). Pyrosequencing: A DNA sequencing method based on real-time pyrophosphate detection. *Science*, 281:363–365.

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

Shchepinov, M. S., Denissenko, M., Smylie, K. J., Wörl, R. J., Leppin, A. L., Cantor, C. R., and Rodi, C. P. (2001). Matrix-induced fragmentation of P3'-N5' phosphoramidate-containing DNA: high-throughput MALDI-TOF analysis of genomic sequence polymorphisms. *Nucleic Acids Res.*, 29(18):3864–3872.

Skiena, S., Smith, W. D., and Lemke, P. (1990). Reconstructing sets from interpoint distances. In *Proceedings of Annual symposium Computational geometry*, pages 332–339.

Skiena, S. S. and Sundaram, G. (1994). A partial digest approach to restriction site mapping. *Bulletin of Mathematical Biology*, 56:275–294.

von Wintzingerode, F., Böcker, S., Schlötelburg, C., Chiu, N. H., Storm, N., Jurinke, C., Cantor, C. R., Göbel, U. B., and van den Boom, D. (2002). Base-specific fragmentation of amplified 16S rRNA genes and mass spectrometry analysis: A novel tool for rapid bacterial identification. *Proc. Natl. Acad. Sci. USA*, 99(10):7039–7044.

Waterman, M. S. (1995). *Introduction to Computational Biology: Maps, sequences and genomes*. Chapman & Hall–CRC Press.

AG Genominformatik, Technische Fakultät, Universität Bielefeld, PF 100 131, 33501 Bielefeld, Germany

E-mail address: boecker@CeBiTec.uni-bielefeld.de

Bisher erschienene Reports an der Technischen Fakultät
Stand: 2003-06-18

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

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

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

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

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

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

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

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

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

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

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

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

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

96-03 Correctness in System Engineering
(Peter Ladkin)

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

96-05 Future University Computing Resources
(Peter Ladkin)

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

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

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

96-09 Reasons and Causes
(Peter Ladkin)

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

96-11 On Needing Models
(Peter Ladkin)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2000-01 Advances in DFS Theory
(Ingo Glöckner)

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

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

2000-04 Affix Trees
(Jens Stoye)

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

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

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

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

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

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

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

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

2001-05 Hierarchical Termination Revisited
(Enno Ohlebusch)

2002-01 Persistent Objects with O2DBI
(Jörn Clausen)

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

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

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

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

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

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

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

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

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

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