Universität Bielefeld

Technische Fakultät
Abteilung Informationstechnik
Forschungsberichte

SNP and mutation discovery using base-specific cleavage and MALDI-TOF mass spectrometry

Sebastian Böcker

Report 2003-02

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

SNP AND MUTATION DISCOVERY USING BASE-SPECIFIC CLEAVAGE AND MALDI-TOF MASS SPECTROMETRY

SEBASTIAN BÖCKER

Abstract. **Motivation:** Single Nucleotide Polymorphisms (SNPs) are believed to contribute strongly to the genetic variability in living beings, in particular their disease or drug side effect predispositions. Mutation-induced sequence variations are playing an important role in the development of cancer, among others. From this, it is clear that SNP and mutation discovery is of great interest in today's Life Sciences. Currently, such discovery is often performed utilizing electrophoresis-based Sanger Sequencing. Discovery of SNPs can also be performed by multiple sequence alignment of publicly available sequence data, but recent studies indicate that only a small percentage of SNPs can be discovered using this approach and, in particular, that SNPs with low frequency are often missed. Other SNP discovery methods only indicate the presence of a SNP in a sample region, but fail to resolve its characterization and localization.

Results: We present a method to discover mutations and SNPs using *base-specific cleavage* and *mass spectrometry*. An amplicon of known reference sequence with length usually between 100 and 1 000 nt is amplified, transcribed, and cleaved using base-specific endonucleases such as RNAse A or T1. The resulting cleavage products (or fragments) are analyzed by MALDI-TOF mass spectrometry and, comparing the measured spectra with those predicted *in-silico*, the goal is to discover and pinpoint sequence variations of the sample sequence compared to the reference sequence. A time-efficient algorithm for discovering sequence variations is presented that enables fast analysis of such variations even if the sample sequence differs significantly from the reference sequence.

Contact: boecker@CeBiTec.uni-bielefeld.de

Keywords: SNPs, molecular sequence analysis, combinatorics, string algorithms

Date: April 25, 2003.

Currently at AG Genome Informatics, Technische Fakultät, Universität Bielefeld, Germany.

1. Introduction

While large parts of an organism's genome are constant across all individuals of a population, there are certain positions in the genome where two or more alternative bases can be observed in a population, and even in the alleles of a single (diploid or polyploid) individual. These polymorphic bases are called SNPs (Single Nucleotide Polymorphisms), and they are widely believed to play an important role in areas such as disease predisposition, drug side effect predisposition, or quantitative and qualitative trait loci in livestock. On the other hand, there are certain deviations in an organism's genome that are observed only in certain cells or cell types of an individual, or in one or a small number of individuals consisting of only a tiny fraction of the overall population. Such deviations are called mutations, and the presence of certain mutations in a cell line is believed to play an important role, for example, in the development of cancer.

A large fraction of today's SNP and mutation discovery is still based on de-novo sequencing of the sample sequences of interest, using the Sanger concept (Sanger et al., 1977) in combination with gel or capillary electrophoresis to acquire the experimental data. This process is comparatively time consuming and does not make use of the sequence information known up-front in such studies. Also, detection of heterozygous SNPs can be difficult. Another, purely computational approach of discovering SNPs is based on sequence alignment of publicly available sequence information like expressed sequence tags (ESTs), an example being SNPpipeline (Buetow et al., 1999). But such approaches seem to find only a small fraction of SNPs present in the genomic region under consideration (Cox et al., 2001). Other approaches like Denaturing Gradient High Pressure Liquid Chromatography (DG-HPLC) or Temperature Gradient Capillary Electrophoresis (TGCE) will only reveal information on the presence or absence of *any* sequence changes in the analyzed region of the sample target, making it necessary to perform a subsequent characterization and localization of the sequence change.

A novel approach for SNP and mutation discovery uses base-specific cleavage of DNA or RNA, and MALDI-TOF mass spectrometry to acquire the experimental data, see (Rodi et al., 2002) for an introduction to the utilized biochemistry and mass spectrometry. This approach is not based on the Sanger concept and has been applied successfully to the problem of bacteria typing (von Wintzingerode et al., 2002).

2. Experimental setup and data acquisition

Suppose we are given a target DNA molecule (or *sample DNA*) of length usually between 100 and 1 000 nt. Using polymerase chain reaction (PCR) or other amplification methods we multiply the sample DNA. We assume that we have a way of generating a single stranded target (either by transcription or other methods), and we will refer to sample DNA even though the cleavage reaction might force us to transcribe the sample to RNA. We cleave the single stranded DNA with a base-specific (bio-)chemical cleavage reaction: Such reactions cleave the amplicon sequence at exactly those positions where a specific base can be found. For example, amplification by PCR and transcription of the product, and subsequent fragmentation using the endonuclease RNAse T1 will cleave the sample sequence wherever rGTP was incorporated, see (Hartmer et al., 2003). Such base-specific cleavage can also be achieved by the use of the endonuclease RNAse A, uracil-DNA-glycosylase (UDG), pn-bond cleavage, and others. In particular, RNAse A can be used to achieve base-specific cleavage of either base C or T, and by transcribing either forward or reverse strand of the sample DNA, this enables us to perform cleavage reactions specific to all four bases (Stanssens et al., 2003).

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

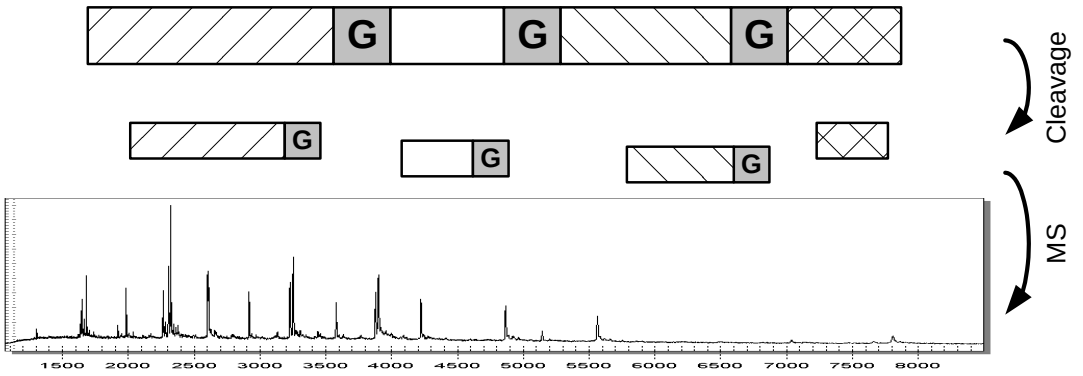

FIGURE 1. Base-specific cleavage using RNAse T1 and subsequent mass spectrometry measurement.

(Karas and Hillenkamp, 1988).[1] The sample spectrum is analyzed to extract a list of signal peaks (with masses and intensities).

We can repeat the above 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 end up with one to four mass spectra, each corresponding to a base-specific cleavage reaction. We repeat the following steps of the analysis for every cleavage reaction.

The exact chemical results of the utilized 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 measured spectra often differ significantly from the *in-silico* predicted spectrum. But compared to collision induced fragmentation in MS/MS mass spectrometry used for peptide de-novo sequencing, there is only a comparatively small number of differences between the simulated spectrum and the measured one. For the approach presented in this paper, measured and predicted peak intensities (peak height, signal-to-noise) in general differ quite strongly. In particular, some fragments present in the sample generate peaks of such small intensity in the measured mass spectrum, that it is impossible to differentiate these peaks from (chemical, physical, and other) "noise" in the mass spectrum. So at present, neither peak intensities nor missing peaks (present in the predicted spectrum but missing from the sample spectrum) but only *additional peaks* (present in the sample spectrum but not expected in the reference spectrum) may be seen as a reliable indicator of sequence variations in the sample sequence. Note further that heterozygous sequence variations never result in missing signals.

A trivial approach to discover[2] sequence variations in a sample would be to simulate the mass spectra for all potential sequence variations of the reference sequence, and to compare the resulting simulated spectra against the measured mass spectrum, finding the one that gives a "best fit" of the measured spectrum. This is in fact a valid approach and can be computed reasonably fast if we limit the potential sequence variations to a single-base substitution, insertion, or deletion. But if we assume that the sample sequence differs at two or more close positions from the reference sequence, this approach

[1]More precisely, MALDI mass spectrometers measure "mass per charge" instead of "mass" of sample particles. For the sake of brevity, we will 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.

[2]We will use the term "discovery" to distinguish this approach from SNP or mutation "detection" where one wants to test whether some previously known SNP or mutation is present in a certain sample.

is not suited for high-throughput analysis. See Section 3.3 for a justification as well as an explanation of the term "close" in this context.

What fragments can account for an additional peak in the sample spectrum? Obviously, we cannot reconstruct the order of bases in a fragment from its mass alone. But for every additional peak in the sample 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: Using MS, masses can be measured up to some uncertainty only. Current ATOF (Axial Time Of Flight) mass spectrometers show uncertainties of 2 or more Dalton[3] under high throughput conditions using nano preparation. This uncertainty can be significantly higher, depending on sample preparation and the mass spectrometer, and special care has to be taken at this point so that mass uncertainties do not become too large.

For a fixed cleavage reaction, several potential base compositions can have nearly identical masses. For example, the DNA molecule `5'OH`-CCGGG-`3'OH` (and, hence, the base composition with 2 C's and 3 G's) has a mass of 1504.0 Da under the natural isotopic distribution; the DNA molecule `5'OH`-AAAAA-`3'OH` has a mass of 1504.1 Da, the mass difference being less than 0.1 Da. But in case of complete base-specific cleavage of DNA, any fragment contains at most 3 out of the 4 bases while the fourth base gets cleaved, and there exist only a limited number of base compositions with mass difference of up to 2 Da (Pomerantz et al., 1993). In the following, we independently use every potential explanation of an additional signal as a base composition. Therefore, we end up with a list of base compositions (with masses sufficiently close to an additional signal in the sample spectrum) depending on the sample DNA and the incorporated cleavage method.

Note that current MALDI-TOF mass spectrometers also limit the mass range in which particles can be detected. A typical mass range of 1 000 to 15 000 Da corresponds to a maximal fragment length of approximately 50 nt that can be detected, but even signals above 8 000 Da (approximately 25 nt) tend to get lost in the spectrum. Simulations indicate that not this limitation, but the incapacity to accurately measure the multiplicity of base compositions in the mass spectrum is responsible for most of the SNPs and mutations we *cannot* discover (Stanssens et al., 2003). In this paper, we will focus on those sequence variations that *can* be discovered by the presented method.

Since MALDI-TOF mass spectrometry reads can be obtained in (milli-)seconds compared to hours for electrophoresis reads, and mass spectrometry generally provides reliable and reproducible results even under high throughput conditions, this seems to be a promising approach for SNP and mutation discovery. But the high throughput of mass spectrometry also renders it necessary to analyze the generated data in a time-efficient way.

Finally, we want to remark that the described approach is closely related to peptide "sequencing" via trypsin digestion, see for example (Mann et al., 1993; Yates III, 1998). But as we will point out below, the problems encountered analyzing data from the above approach differ strongly from those of peptide "sequencing."

3. Methods

3.1. **Strings and string spectra.** Let $s \in \Sigma^*$ denote a string over the alphabet Σ where $|s|$ denotes the *length* of s. The concatenation of strings a, b will be denoted ab, the empty string of length 0 is also denoted ϵ.

If $s = axb$ holds for some strings a, x, b then x is called a *substring* of s, denoted $x \preceq s$. If $s = ab$ holds for some strings a, b then a is called a *prefix* of s, and b is called an *suffix*

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

of s. For $1 \leq i \leq j \leq |s|$ we will denote the substring $s_i s_{i+1} \dots s_{j-1} s_j$ of $s = s_1 \dots s_{|s|}$ by $s_{[i,j]}$.

Given strings s and x from Σ^*, we define the *string spectrum* of s with respect to x, denoted $\mathcal{S}_0(s,x)$, by:

$$\begin{aligned} \mathcal{S}_0(s,x) := \big\{ y \in \Sigma^* : \text{ there exist } a,b \in \Sigma^* \\ \text{with } s \in \{yxb, axyxb, axy\}, \text{ and } x \not\preceq y \big\} \end{aligned} \tag{1}$$

The string spectrum $\mathcal{S}_0(s,x)$ consists of those substrings of s that are "bounded" by x (or the ends of s), and that do not contain x. In this context, s will be called *sample string* or *reference string*, x will be called *cut string*, while the elements of $\mathcal{S}_0(s,x)$ will be called *fragments* of s (under x).

Example 1. Let $s :=$ ACATGTGCCATTA and $x :=$ T over the alphabet $\Sigma := \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}, \}$, then:

$$\mathcal{S}_0(s,x) = \{\mathrm{ACA}, \mathrm{G}, \mathrm{GCCA}, \epsilon, \mathrm{A}\}$$

The *order* of a string s with respect to x, denoted $\mathrm{ord}_x(s)$ or $\mathrm{ord}(s)$ for short, counts the maximal number of times the string s can be cleaved under x:

$$\begin{aligned} \mathrm{ord}_x(s) := \max\big\{ k \in \mathbb{N} : \text{ there exist } y_0, \dots, y_k \in \Sigma^* \\ \text{such that } s = y_0 x y_1 x \dots x y_{k-1} x y_k \big\} \end{aligned}$$

Clearly, $\mathrm{ord}_x(y) = 0$ holds for all $y \in \mathcal{S}_0(s,x)$, what justifies the notation $\mathcal{S}_0(s,x)$. For cut strings x whose only period is x itself (Apostolico and Galil, 1997) and, in particular, for cut strings of length one, $\mathrm{ord}_x(s)$ simply counts how often x appears as a substring of s. But the above is not true for arbitrary strings s, x as the simple example $s :=$ AAA and $x :=$ AA shows. For fixed x, $\mathrm{ord}_x(s)$ can be computed in $O(|s|)$ time by greedily searching for the next appearance of x in s (Boyer and Moore, 1977; Knuth et al., 1977).

We denote the edit distance with unit cost (Levenshtein distance) between two strings s, s' by $d_{\mathrm{L}}(s,s')$. This is the number of insertions, deletions, and substitutions minimally needed to transform s into s'. We will use the term *sequence variation* to describe the transformation of some string s into another string s' via a sequence alignment. We will use the terms "sequence variation" and "alignment" interchangeably. The cost of a sequence variation is the cost of the corresponding alignment.

3.2. **Compomers and compomer spectra.** As 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), and let $\mathcal{C}(\Sigma)$ denote the set of all compomers over the alphabet Σ. Clearly, $\mathcal{C}(\Sigma)$ forms a group that is closed with respect to multiplications with a scalar $n \in \mathbb{Z}$. For finite Σ, in particular, $\mathcal{C}(\Sigma)$ is isomorphic to the set $\mathbb{Z}^{|\Sigma|}$. We denote the canonical partial order on $\mathcal{C}(\Sigma)$ by $\preceq$, that is, $c \preceq c'$ iff $c(\sigma) \leq c'(\sigma)$ for all $\sigma \in \Sigma$. We will denote the *empty compomer* $c \equiv 0$ by 0. Finally, we define the *absolute value* of c by $|c| := \sum_{\sigma \in \Sigma} |c(\sigma)|$.

Given a compomer c over Σ we use the notations $c_{\geq 0}, c_{\leq 0}$ for those compomers in $\mathcal{C}(\Sigma)$ defined by:

$$\begin{aligned} c_{\geq 0}(\sigma) &:= \max\{c(\sigma), 0\} \\ c_{\leq 0}(\sigma) &:= \min\{c(\sigma), 0\} \end{aligned} \quad \text{for } \sigma \in \Sigma\,. \tag{2}$$

Clearly, $c = c_{\geq 0} + c_{\leq 0}$. We say that a compomer c is a *natural compomer* if $c = c_{\geq 0}$ holds. The set of natural compomers over Σ is closed with respect to addition as well as multiplication with scalars $n \in \mathbb{N}$, where $\mathbb{N}$ denotes the set of natural numbers *including* 0.

Suppose that $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$, then we use the notation $(\sigma_1)_{i_1} \ldots (\sigma_k)_{i_k}$ to represent the compomer $c : \sigma_j \mapsto i_j$ omitting those letters σ_j with $i_j = 0$. In case of 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 $c(\mathrm{A}) = i$, $\ldots$, $c(\mathrm{T}) = l$.

We define the function $\mathrm{comp} : \Sigma^* \to \mathcal{C}(\Sigma)$ such that a string $s = s_1 \ldots s_n \in \Sigma^*$ is mapped to the compomer counting the number of letters in s:

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

Obviously, $\mathrm{comp}(s)$ is a natural compomer. Note that compomers $\mathrm{comp}(\cdot)$ are also referred to as Parikh-vectors, see (Autebert et al., 1997). The *compomer spectrum* $\mathcal{C}_0(s, x)$ of s consists of the compomers of all fragments in the string spectrum:

$$\begin{aligned} \mathcal{C}_0(s,x) :=& \mathrm{comp}\left(\mathcal{S}_0(s,x)\right) \\ =& \big\{\mathrm{comp}(y) \,:\, y \in \mathcal{S}_0(s,x)\big\} \end{aligned} \tag{3}$$

For Example 1 we can calculate $\mathcal{C}_0(s, \mathrm{T}) = \{\mathrm{A}_2\mathrm{C}_1,\ \mathrm{G}_1,\ \mathrm{A}_1\mathrm{C}_2\mathrm{G}_1,\ 0,\ \mathrm{A}_1\}$.

Given two compomers c, c' over Σ corresponding to (unknown) fragment strings y, y', how many insertions, deletions, and substitutions will it *minimally* take to transform y into y'? As an example, suppose $c = \mathrm{A}_1\mathrm{C}_1$ and $c' = \mathrm{C}_1\mathrm{G}_1$, then y may equal AC or CA, and y' may equal CG or GC. We calculate $d_{\mathrm{L}}(\mathrm{AC}, \mathrm{CG}) = d_{\mathrm{L}}(\mathrm{CA}, \mathrm{GC}) = 2$ and $d_{\mathrm{L}}(\mathrm{AC}, \mathrm{GC}) = d_{\mathrm{L}}(\mathrm{CA}, \mathrm{CG}) = 1$, so the answer equals one for this example. It follows from the construction presented in the next paragraph that this number equals $d(c, c')$ where $d : \mathcal{C}(\Sigma) \times \mathcal{C}(\Sigma) \to \mathbb{N}$ is defined by

$$d(c,c') := \max\Big\{\big|(c-c')_{\ge 0}\big|, \big|(c-c')_{\le 0}\big|\Big\}. \tag{4}$$

The function d forms a metric on $\mathcal{C}(\Sigma)$, where the triangle inequality follows directly from $\big|c_{\ge 0}\big| + \big|c'_{\ge 0}\big| \ge \big|(c+c')_{\ge 0}\big|$ as well as $\big|c_{\le 0}\big| + \big|c'_{\le 0}\big| \ge \big|(c+c')_{\le 0}\big|$. From the above, we conclude $d_{\mathrm{L}}(y, y') \ge d(c, c')$.

To this end, suppose that we are given a fragment $y \in \Sigma^*$ and a compomer $c' \in \mathcal{C}(\Sigma)$. We define $c := \mathrm{comp}(y)$, and we want to construct *all* fragments $y' \in \Sigma^*$ satisfying $\mathrm{comp}(y') = c'$ such that $d_{\mathrm{L}}(y, y') = d(c, c')$ holds. The idea of the following algorithm is to use exactly $d(c, c')$ substitutions, insertions, and deletions, one in every step of the recursion, to transform y into some fragment y'. Formally, we can show by induction on $|c - c'|$ that every such fragment y' can be constructed by the following recursion:

1. We assume that $y = y_1 \ldots y_n$. Set $\Delta := c' - c \in \mathcal{C}(\Sigma)$, $k := d(c, c')$, $k_+ := |\Delta_{\ge 0}|$, and $k_- := |\Delta_{\le 0}|$. By definition, $k = \max\{k_+, k_-\}$ holds.
2. Let $sub := \min\{k_+, k_-\}$, $ins := k_+ - sub$, and $del := k_- - sub$. Clearly, $sub + ins + del = k$.
3. Finally, set $\Sigma_+ := \{\sigma \in \Sigma \,:\, \Delta(\sigma) > 0\}$ and $\Sigma_- := \{\sigma \in \Sigma \,:\, \Delta(\sigma) < 0\}$, and note that $\Sigma_+ \cap \Sigma_- = \emptyset$.
4. For $\Delta = 0$ return y.
5. Otherwise, do *one* of the following as the recursion step:
   - If $sub > 0$, then choose an index i with $1 \le i \le n$ satisfying $y_i \in \Sigma_-$, and a letter $\sigma' \in \Sigma_+$. Do the recursion with $y' := y_1 \ldots y_{i-1}\sigma' y_{i+1} \ldots y_n$.
   - If $del > 0$, then choose an index i with $1 \le i \le n$ satisfying $y_i \in \Sigma_-$. Do the recursion with $y' := y_1 \ldots y_{i-1} y_{i+1} \ldots y_n$.
   - If $ins > 0$, then choose an index i with $1 \le i \le n+1$, and a letter $\sigma' \in \Sigma_+$. Do the recursion with $y' := y_1 \ldots y_{i-1}\sigma' y_i \ldots y_n$.

We repeat the above recursion until $\Delta = c' - c$ equals 0. If we let the index i run from 1 to $n + 1$ (not taking into account index shifts due to insertions and deletions) we can ensure that *every* admissible fragment y' (resp. the corresponding sequence variation) gets constructed exactly once.

We will now give a formal representation of the problem of SNP and mutation discovery. Suppose we are given a reference string $s \in \Sigma^*$ and a cut string $x \in \Sigma^*$. From the measured mass spectrum, we have constructed a set of compomers $\mathcal{C} \subseteq \mathcal{C}(\Sigma)$. In case of a homozygous sample, this set $\mathcal{C}$ corresponds to the compomer spectrum $\mathcal{C}_0(s', x)$ of the unknown sample string s', while for the case of a heterozygous sample, $\mathcal{C}$ corresponds to the union of compomer spectra $\mathcal{C}_0(s', x) \cup \mathcal{C}_0(s'', x)$ where s', s'' are the alleles of the sample. We assume $s'' = s$, i.e. we know at least one allele of a heterozygous sample in advance.

As we have explained above, we cannot fully rely on signal intensities or missing signals in the sample spectrum, in particular the latter, in view of potentially heterozygous samples. Hence, we define:

SNP Discovery from Mass Spectrometry Problem. Given a reference string $s \in \Sigma^*$ and a cut string $x \in \Sigma^*$. For a compomer $c' \in \mathcal{C}(\Sigma)$ find all $s' \in \Sigma^*$ satisfying $c' \in \mathcal{C}_0(s', x)$ such that $d_{\mathrm{L}}(s, s')$ is minimal.

The minimality criterion is motivated from the goal of parsimony, that is, maximizing string similarity or minimizing (evolutionary) distance much like in case of sequence alignments.

3.3. **Independent sequence variations.** Suppose we are given a random i.i.d. string $s = s_1 s_2 s_3 \ldots$ on an alphabet of four letters. Let the random variable X denote the natural number so that the substring $s' = s_1 \ldots s_X$ contains exactly three letters of the alphabet, while $s'' = s_1 \ldots s_{X+1}$ contains all four letters. The expected length of such a substring is $E(X) = 7\frac{1}{3}$, see Lemma 2 in the Appendix.

This helps us to roughly estimate how often two sequence variations are *independent*: Informally, this means that the changes in the compomer spectra resulting from one sequence variations, and the changes in the compomer spectra resulting from a second sequence variation simply add up to form the changes in compomer spectra of the composite sequence variation. To illustrate this, let $s :=$ AGCCTGTT, and suppose the sequence variations under examination are substitution C/A at position 3 of s, and substitution T/C at position 5 of s. Let $x :=$ G denote the cut string, then these sequence variations are dependent with respect to x: We calculate $\mathcal{C}_0(s, x) = \{\mathrm{A}_1, \mathrm{C}_2\mathrm{T}_1, \mathrm{T}_2\}$, $\mathcal{C}_0(s_1, x) = \{\mathrm{A}_1, \mathrm{A}_1\mathrm{C}_1\mathrm{T}_1, \mathrm{T}_2\}$ for $s_1 =$ AGACTGTT, $\mathcal{C}_0(s_2, x) = \{\mathrm{A}_1, \mathrm{C}_3, \mathrm{T}_2\}$ for $s_2 =$ AGCCCGTT, and $\mathcal{C}_0(s_{1,2}, x) = \{\mathrm{A}_1, \mathrm{A}_1\mathrm{C}_2, \mathrm{T}_2\}$ for $s_{1,2} =$ AGACCGTT. Now, $\mathcal{C}_0(s_{1,2}, x) \not\subseteq \mathcal{C}_0(s_1, x) \cup \mathcal{C}_0(s_2, x)$ implies that these sequence variations cannot be independent with respect to $x =$ G. On the other hand, the above sequence variations are independent with respect to $x =$ C, because the letter 'C' at position 4 of s "divides" the two sequence variations. In total, the sequence variations are dependent because there exists (at least) one cut string under consideration such that the sequence variations are dependent with respect to this cut string.

Clearly, two sequence variations are independent if the substring between them contains all letters of the alphabet, in case all cut strings have length one. This means that even when limiting ourselves to comparatively small sequence variation costs k we can often reconstruct sample sequences s' with $d_{\mathrm{L}}(s, s') \gg k$. For example, SNPs are rather sparsely distributed across the human genome: For the SNP discovery study described in Section 5, we analyzed 11793 base pairs and discovered 51 SNPs (one SNP every 231 base pairs on average), the minimal distance between any two discovered SNPs being 14 base pairs. This indicates that the desirable threshold to be reached in case of SNP Discovery equals $k = 1$ or $k = 2$, the latter covering the rare cases where two SNPs are in close vicinity. In case of Mutation Discovery, multiple base changes in close vicinity are more frequently observed, so a sequence variation cost $k = 3$ or $k = 4$ might be useful for this application type.

3.4. **Boundaries and bounded compomers.** When we compare the compomer spectra of two similar strings $s, s' \in \Sigma^*$, it is clear that $\mathcal{C}_0(s,x)$ and $\mathcal{C}_0(s',x)$ may differ significantly if we allow the insertion of or substitution to (parts of) the cut string x, in which case compomers in $\mathcal{C}_0(s,x)$ might be "cleaved" into two or more compomers in $\mathcal{C}_0(s',x)$. Similarly, the deletion or substitution of (parts of) the cut string x might "merge" compomers of the source spectrum. Unlike so-called "de-novo" sequencing of peptides via trypsin digestion (Mann et al., 1993; Yates III, 1998), these events are not exceptions but the rule in our approach.

To address this problem, we will not only keep track of the compomer induced by some fragment y of a string s, but also whether it is necessary to "insert" cut strings before and after the fragment in s. Formally, we define a *boundary* b to be an element of the set $\mathcal{B} := \mathcal{P}(\{\mathrm{L},\mathrm{R}\}) = \big\{\emptyset, \{\mathrm{L}\}, \{\mathrm{R}\}, \{\mathrm{L},\mathrm{R}\}\big\}$. A *bounded compomer* (c,b) over Σ consists of a compomer $c \in \mathcal{C}(\Sigma)$ and a boundary $b \in \mathcal{B}$. We define the distance measure $D : \mathcal{C}(\Sigma) \times \mathcal{B} \times \mathcal{C}(\Sigma) \to \mathbb{N}$ by

$$D(c,b,c') := d(c,c') + |b| \ . \tag{5}$$

The function D does not only count the number of transformations needed to transform fragment y into y' (corresponding to the compomers c and c', respectively) but also adds 0,1, or 2 to account for the number of transformations needed to obtain y as an element of the string spectrum of the underlying reference string s.

Given a sample string $s \in \Sigma^*$ and a cut string $x \in \Sigma^*$, we say that s is *left-bounded* at index i, $1 \le i \le |s|$, if either $i = 1$ holds, or if x is a suffix of $s_{[1,i-1]}$. Analogously, we say that s is *right-bounded* at index j, $1 \le j \le |s|$, if either $j = |s|$ holds, or if x is a prefix of $s_{[i+1,|s|]}$. We define the function $b_{s,x} : [1,|s|]^2 \to \mathcal{B}$ by

$$\begin{aligned} b_{s,x}(i,j) :=& \{\mathrm{L} \,:\, s \text{ is not left-bounded at } i\} \\ & \cup \{\mathrm{R} \,:\, s \text{ is not right-bounded at } j\} \end{aligned} \tag{6}$$

So, $\mathrm{L} \in b_{s,x}(i,j)$ holds if s is not cleaved before index i, and therefore we have to modify the string s to achieve such cleavage; and $\mathrm{R} \in b_{s,x}(i,j)$ holds if s is not cleaved after index j. We define:

$$\begin{aligned} \mathcal{S}^{\mathrm{B}}(s,x) &:= \Big\{ \big(s_{[i,j]}, b_{s,x}(i,j)\big) \,:\, 1 \le i \le j \le |s| \Big\} \\ \mathcal{C}^{\mathrm{B}}(s,x) &:= \big\{ (\mathrm{comp}(y), b) \,:\, (y,b) \in \mathcal{S}^{\mathrm{B}}(s,x) \big\} \end{aligned}$$

as the set of bounded strings (compomers) of s, x. Clearly, $c \in \mathcal{C}_0(s,x)$ iff $(c,\emptyset) \in \mathcal{C}^{\mathrm{B}}(s,x)$. The set $\mathcal{C}^{\mathrm{B}}(s,x)$ contains all those fragments that can be obtained from s by "inserting" cut strings x at appropriate positions.

Example 2. For the sample string $s :=$ ATTCA and $x :=$ T we calculate $\mathcal{S}^{\mathrm{B}}(s,x) = \big\{(\mathrm{A},\emptyset),\ (\mathrm{T},\mathrm{L}),\ (\mathrm{T},\mathrm{R}),\ (\mathrm{C},\mathrm{R}),\ (\mathrm{A},\mathrm{L}),\ (\mathrm{AT},\emptyset),\ (\mathrm{TT},\mathrm{LR}),\ (\mathrm{TC},\mathrm{R}),\ (\mathrm{CA},\emptyset),\ (\mathrm{ATT},\mathrm{R}),\ (\mathrm{TTC},\mathrm{LR}),\ (\mathrm{TCA},\emptyset),\ (\mathrm{ATTC},\mathrm{R}),\ (\mathrm{TTCA},\mathrm{L}),\ (\mathrm{ATTCA},\emptyset)\big\}$.

We will not use the set $\mathcal{C}^{\mathrm{B}}(s,x)$ directly, but instead define a subset for a more restrictive claim in Theorem 1 below. To this end, we define for $k \in \mathbb{N}$:

$$\begin{aligned} \mathcal{C}^{\mathrm{B}}_k(s,x) := \big\{ (\mathrm{comp}(y), b) \,:\, & (y,b) \in \mathcal{S}^{\mathrm{B}}(s,x) \\ & \text{and } \ \mathrm{ord}_x(y) + |b| \le k \big\} \end{aligned} \tag{7}$$

For Example 2 we calculate $\mathcal{C}^{\mathrm{B}}_1(s,x) = \big\{(\mathrm{A}_1,\emptyset),\ (\mathrm{C}_1,\mathrm{R}),\ (\mathrm{A}_1,\mathrm{L}),\ (\mathrm{A}_1\mathrm{T}_1,\emptyset),\ (\mathrm{A}_1\mathrm{C}_1,\emptyset),\ (\mathrm{A}_1\mathrm{C}_1\mathrm{T}_1,\emptyset)\big\}$.

Theorem 1. *Given sample strings $s, s' \in \Sigma^*$ with $d_{\mathrm{L}}(s,s') \le k$, a cut string $x \in \Sigma^*$, and a compomer $c' \in \mathcal{C}_0(s',x)$. Then, there exists a bounded compomer $(c,b) \in \mathcal{C}^{\mathrm{B}}_k(s,x)$ such that $D(c,b,c') \le d_{\mathrm{L}}(s,s')$.*

Proof. Let i', j' denote integers such that $y' := s'_{[i',j']}$ satisfies $\mathrm{comp}(y') = c'$ as well as $s' \in \{y'xb, axy'xb, axy'\}$ for some $a, b \in \Sigma^*$. By definition, $y' \in \mathcal{S}_0(s', x)$. Under a minimal alignment of s and s', let i, j denote integers such that positions $[i, j]$ of s correspond to positions $[i', j']$ of s' in the alignment. Set $y := s_{[i,j]}$, then $d(c, c') \leq d_{\mathrm{L}}(y, y')$ holds for $c := \mathrm{comp}(y)$. Let $b := b_{s,x}(i, j)$, then $(c, b) \in \mathcal{S}^{\mathrm{B}}(s, x)$. From $d_{\mathrm{L}}(y, y') \leq d_{\mathrm{L}}(s, s') - |b|$ we infer

$$D(c, b, c') = d(c, c') + |b| \leq d_{\mathrm{L}}(y, y') + |b| \leq d_{\mathrm{L}}(s, s').$$

It remains to be shown that $\mathrm{ord}_x(y) \leq k - |b|$. But from the definition of $\mathrm{ord}_x(\cdot)$ we can easily infer $d_{\mathrm{L}}(y, y') \geq |\mathrm{ord}_x(y) - \mathrm{ord}_x(y')|$ and, hence,

$$\begin{aligned} k &\geq d_{\mathrm{L}}(s, s') \geq d_{\mathrm{L}}(y, y') + |b| \\ &\geq \left|\mathrm{ord}_x(y) - \mathrm{ord}_x(y')\right| + |b| = \mathrm{ord}_x(y) + |b|\,. \end{aligned}$$

□

Theorem 1 is the building block our algorithm below is based on: Informally, if the unknown sample sequence differs (not too much) from the reference sequence, leading to the detection of an additional compomer c', then there exists one or more bounded compomers (c, b) close to c' (with respect to D). In fact, the bounded compomers (c, b) can be used to construct all "minimal" sample sequence candidates that explain the observed additional compomer c', so all we have to do is to construct the set $\mathcal{C}^{\mathrm{B}}_k(s, x)$, then search if we can find bounded compomers (c, b) close to c'. For cut strings x of length 1, we can guarantee that there exists a sequence variation of minimal cost that will generate an observed additional compomer.

Lemma 1. *Given a reference string $s \in \Sigma^*$ and a cut string $x \in \Sigma^1$ of length 1. For a compomer $c' \in \mathcal{C}(\Sigma)$ and a bounded compomer $(c, b) \in \mathcal{C}^{\mathrm{B}}(s, x)$, there exists a string $s' \in \Sigma^*$ satisfying $d_{\mathrm{L}}(s, s') = D(c, b, c')$ and $c' \in \mathcal{C}_0(s', x)$.*

Proof. Let i, j denote integers such that $\mathrm{comp}(s_{[i,j]}) = c$. We have seen in Section 3.2 how to construct a fragment string $y' \in \Sigma^*$ such that $d_{\mathrm{L}}(s_{[i,j]}, y') = d(c, c')$ and $\mathrm{comp}(y') = c'$. We replace the substring at positions i to j of s by y'. In addition, we apply the following sequence variations: For $\mathrm{L} \in b$ we substitute position $i - 1$ of s by x, or we insert x before position i into s. For $\mathrm{R} \in b$ we substitute position $j + 1$ of s by x, or we insert x after position j. Let s' denote the string resulting from these sequence variation. Then, $d_{\mathrm{L}}(s, s') = d(c, c') + |b| = D(c, b, c')$ and $c' \in \mathcal{C}_0(s', x)$ as required. □

Again, we can show that every string $s' \in \Sigma^*$ satisfying the conditions of the lemma will be generated by the construction presented in the proof of the lemma. But note that the sequence variations generated this way are not necessarily minimal, and that to guarantee minimality we have to take special care when generating the "bounds" of the fragment y'.

Clearly, Lemma 1 does not hold in case $|x| \geq 2$. We can modify our definition of boundaries accordingly: Then, a boundary is a vector $b = (b_1, b_2) \in \mathbb{N}^2$ where b_1 (b_2) counts the number of transformations needed to obtain cleavage on the left (right) side of the fragment, $|b| := b_1 + b_2$, and equation (5) is defined for $\mathcal{B} := \mathbb{N}^2$. But even then, we cannot guarantee the existence of a sequence variation of the observed cost, see Example 3. Still, if a minimal sequence variation exists, it will be generated by the construction presented in the proof of Lemma 1, so a simple post-processing step can be applied to sort out sequence variations not generating the observed compomer.

Example 3. Let $s :=$ TGCT, $x :=$ AC, and $c' := \mathrm{A}_1\mathrm{C}_1\mathrm{T}_2$. Then $\mathcal{C}_0(s, x) = \{\mathrm{C}_1\mathrm{G}_1\mathrm{T}_2\}$ and there exists a bounded compomer $(c, b) \in \mathcal{C}^{\mathrm{B}}(s, x)$ with $D(c, b, c') = 1$, namely $(c, b) = (\mathrm{C}_1\mathrm{G}_1\mathrm{T}_2, \emptyset)$ or $(c, b) = (\mathrm{C}_1\mathrm{G}_1\mathrm{T}_2, (0, 0))$.

But there exists no string $s' \in \Sigma^*$ satisfying $d_{\mathrm{L}}(s, s') = 1$ and $c' \in \mathcal{C}_0(s', x)$ at the same time: Obviously, $c' \preceq \mathrm{comp}(s')$ must hold. The only strings $s' \in \Sigma^*$ satisfying

$d_{\mathrm{L}}(s, s') = 1$ and $c' \preceq \mathrm{comp}(s')$ are $s' =$ TACT, ATGCT, TAGCT, TGACT, TGCAT, and TGCTA. But then, $\mathcal{C}_0(s', x)$ equals $\{\mathrm{T}_1\}$, $\{\mathrm{A}_1\mathrm{C}_1\mathrm{G}_1\mathrm{T}_2\}$, or $\{\mathrm{G}_1\mathrm{T}_1, \mathrm{T}_1\}$.

4. Algorithm

In the following, we present an algorithm to solve the Problem of SNP Discovery from Mass Spectrometry. To improve readability, we limit ourselves to the case where all cut strings have length one. This case is most relevant for applications; the general case can be solved in a comparable fashion.

Let Σ denote the fixed and finite alphabet. Given the reference string $s \in \Sigma^*$ of length $n := |s|$, the cut string $x \in \Sigma^1$ of length one, and a compomer $c' \in \mathcal{C}(\Sigma)$ we want to construct *all* potential sample strings $s' \in \Sigma$ satisfying $c' \in \mathcal{C}_0(s', x)$ such that $d_{\mathrm{L}}(s, s')$ is minimal. In addition, we want to limit this search to a maximal cost k, that is, we also require $d_{\mathrm{L}}(s, s') \leq k$. This limitation is necessary only in order to reduce the number of constructed potential sample strings that have to be scored.

In the preprocessing step, we compute all *indexed bounded compomers* (c, b, i, j) for $1 \leq i \leq j \leq n$ where $c := \mathrm{comp}(s_{[i,j]})$ and $b := b_{s,x}(i, j)$. We store those elements that satisfy $\mathrm{ord}_x(s_{[i,j]}) + |b| \leq k$. This can be done in runtime and memory $O(n^2)$. We sort these elements with respect to $|c|$ what can also be done in runtime $O(n^2)$ in view of $0 \leq |c| \leq n$, and for every $0 \leq m \leq n$ we store indices of the first and last indexed bounded compomer satisfying $|c| = m$ in a lookup table of size $O(n)$. Note that for fixed m, there exist $O(n)$ many elements (c, b, i, j) satisfying $|c| = m$. All of the above bounds are tight for certain reference strings, see Example 4 below.

In the processing step, we are given the "additional compomer" $c' \in \mathcal{C}(\Sigma)$ and want to construct sequence variations leading to sample strings s' with $c' \in \mathcal{C}_0(s', x)$ such that $d_{\mathrm{L}}(s, s') \leq k$ is minimal.

Let k' denote the minimum value of D, which can be found in $O(n \cdot k)$ runtime in case $D(c, b, c') \leq k$: This implies $d(c, c') \leq k$, so we can limit this search to those bounded compomers (c, b) satisfying $|c'| - k \leq |c| \leq |c'| + k$. If we do not find any bounded compomer satisfying $D(c, b, c') \leq k$, then we return the empty set. In the next step, we search for all indexed bounded compomers (c, b, i, j) satisfying $D(c, b, c') = k'$. In view of $|b| \leq 2$ this can be done in $O(n)$ runtime, and there exist $O(n)$ such elements, because $\big||c| - |c'|\big| \geq k'$ implies $D(c, b, c') \geq k'$.

For every (c, b, i, j) satisfying $D(c, b, c') = k'$ we can now use the algorithm of Section 3.2 to construct all fragments $y' \in \Sigma^*$ that satisfy $\mathrm{comp}(y') = c'$ and $d_{\mathrm{L}}(y, y') = d(c, c')$ for $y := s_{[i,j]}$. The complete sequence variation consists of the sequence variation transforming y to y', shifted to index i of s, plus a substitution by or insertion of the letter x at index $i - 1$ in case $\mathrm{L} \in b$, plus a substitution by or insertion of the letter x at index $j + 1$ in case $\mathrm{R} \in b$. Clearly, there exist at most $4 \cdot |\Sigma|^m$ such sequence variations for $m := |c'|$, and this is independent of n and k. In applications, $|c'| \leq 25$ holds as noted in Section 2. But this bound is only of theoretical interest for small k, because $4 \cdot 4^{25} \approx 4.5 \cdot 10^{15}$, and a tighter bound can be found in this case: For the worst case of constructing insertions only, there exist at most $|\Sigma|^k \cdot \binom{m+k}{k}$ sequence variations transforming y to y', so the number of constructed sequence variations is of order $O(|\Sigma|^k \cdot (m + k)^k)$. Using an appropriate branch-and-bound modification of the algorithm of Section 3.2, this can be done in time $O(k \cdot |\Sigma|^k \cdot (m + k)^k)$ taking into account the maximal size k of every constructed sequence variation.

Formally, it remains to be shown that the presented algorithm will construct all sequence variations of minimal cost. This can be done using Theorem 1, but we omit the proof for the sake of brevity. Note that the runtime $O(n \cdot k)$ when finding the minimum of D could be further reduced to $O(n)$ by generating a hash table of all indexed bounded compomers that have distance at most k to a bounded compomer in $\mathcal{C}^{\mathrm{B}}(s, x)$. But this would significantly increase memory requirements and is of no use

in applications, where most of the runtime is spent on scoring the potential sequence variations.

Example 4. For even n, set $s := (\mathrm{AC})^{n/2}$ and $x := \mathrm{G}$. For all $1 \le i \le j \le n$ we define $m := \lfloor \frac{j+1-i}{2} \rfloor$, $b_{i,j} := b_{s,x}(i,j)$, and

$$c_{i,j} := \mathrm{A}_m\mathrm{C}_m + \begin{cases} \mathrm{A}_1 & \text{if } i \text{ and } j \text{ are odd,} \\ \mathrm{C}_1 & \text{if } i \text{ and } j \text{ are even,} \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, $c_{i,j} = \mathrm{comp}(s_{[i,j]})$. Furthermore, $\mathrm{L} \in b_{i,j}$ holds iff $i \neq 1$, and $\mathrm{R} \in b_{i,j}$ holds iff $j \neq n$. Then,

$$\mathcal{C}_2^{\mathrm{B}}(s,x) = \big\{(c_{i,j}, b_{i,j}, i, j) \,:\, 1 \le i \le j \le n\big\}$$

and $\big|\mathcal{C}_2^{\mathrm{B}}(s,x)\big| = \binom{n+1}{2}$. Note further that for every $m = 0, \ldots, n$, there exist $(n+1-m)$ elements satisfying $|c_{i,j}| = m$.

The complete process of discovering sequence variations can now be performed as follows: For every cleavage reaction used, we compute the string and compomer spectrum, and use these to create a reference mass spectrum. We compare the simulated reference mass spectrum to the measured sample mass spectrum, and identify additional peaks in the measured spectrum. For ever additional peak in the measured spectrum, we calculate all compomers with mass sufficiently close to that of the additional peak. For every such compomer, we use the algorithm presented in this section to compute all potential sequence variation leading to the generation of the compomer.

We thereby generate a set of potential sequence variations that must be evaluated taking into account the mass spectrometry data available from *all* cleavage reactions. Given a sample sequence candidate s' we simulate, for every cleavage reaction, the mass spectrum of s' and compare it with the measured one. A very simple scoring scheme is as follows: Let M, M' denote the simulated mass spectra of s and s'. We use the differences of M and M' to calculate two scores, a heterozygous score f_{het} and a homozygous score f_{hom}, both initialized to zero.

- Let p' denote a peak in $M' \setminus M$. If p' is found in the measured mass spectrum, then we add $+1$ to f_{het} and f_{hom}. If p' is not found in the measured mass spectrum, then we add -1 to f_{het} and f_{hom}.
- Let p denote a peak in $M \setminus M'$. If p is not found in the measured mass spectrum, then we add $+1$ to f_{hom}. If p is found in the measured mass spectrum, then we add -1 to f_{hom}.

The overall score of the sequence variation candidate is the maximum of f_{het} and f_{hom}. It is obvious that the presented scheme can easily be refined, for example by taking into account peak intensities. For the results presented in the next section, the scoring scheme was refined by (a) slightly decreasing the score of "false positive peaks" (last item listed above) to -1.2; (b) weighting a peak's score based on the overall quality of the underlying mass spectrum; (c) weighting a peak's score based on its mass, since false positive/negative peaks appear more regularly in the lower mass range of a mass spectrum; and (d) scoring peak intensity variations, where an intensity variation can modify the overall scores by some value in the range $[-0.5, 0.5]$.

As a mathematical justification of the presented scheme, we want to point out that it resembles a maximum likelihood approach, summing the log likelihoods when comparing the model [reference sequence is s'] to the model [reference sequence is s]. A more advanced scoring scheme could compute these likelihoods to calculate a score for a potential sequence variation.

5. Results

The presented algorithms were implemented in C++ as part of the proprietary MassARRAY^{TM} SNP Discovery package from SEQUENOM, Inc. Included in this implementation is a refined SNP scoring scheme as suggested in the previous section, and an iterative SNP selection process. It has already been used to analyze several genomic regions.

Because there exists no "gold standard" data to evaluate the performance of the presented method, we compared it to results obtained from manual analysis of mass spectrometry data, and to classical Sanger Sequencing using electrophoresis data. As an example, we will describe in the following the performance of the presented method analyzing regions on Human Chromosome 22. On this chromosome, 30 "disjoint" amplicons (non-overlapping sub-regions of DNA amplified by PCR) of lengths 328 to 790 base pairs were amplified, the average length of an amplicon being 433 base pairs. In total, 11793 base pairs were analyzed; DNA samples from 12 Caucasian individuals were used (Dausset et al., 1990). For the mass spectrometric analysis, four base-specific cleavage reactions were performed using RNAse A and measured by mass spectrometry independently. All experiments were performed single-pass without repetition, and bad spectra (due to failed biochemical reactions) were not removed from the analysis.

Analyzing the mass spectrometry data *manually*, 50 SNPs were discovered and verified by an independent method (chain terminating primer extension). For 6 of these 50 SNPs, the exact position could not be determined from the cleavage mass spectrometry data. Manual analysis of the mass spectrometry data was very time consuming, and it took several weeks to complete the analysis. Analysis of the *electrophoresis data* indicated that at least 4 of the 50 SNPs would have been missed without manual inspection of sequencing traces. In addition, one SNP was found using the electrophoresis data that was missed in the manual analysis of the mass spectrometry data.

In total, 51 SNPs were discovered by manual analysis of mass spectrometry data or electrophoresis data. 36 of these 51 SNPs (71 %) were previously known and available in public databases, 15 SNPs (29 %) were novel. Considering that the Human Chromosome 22 is a well-studied genomic region, it is noteworthy that almost one third of the discovered SNPs were missing from public databases. In the following, we will assume that these 51 SNPs are exactly the true positives of the test set.

The cleavage mass spectrometry data was then analyzed without user interaction using the automated SNP discovery package as outlined in Section 4, including the presented algorithm to construct sequence variations with maximal cost $k = 1$. All of the 51 SNPs were included in the 22 447 potential sequence variations constructed using the presented algorithm. The analysis was performed for every sample individually, so 1871 sequence variations per sample were scored on average. Subsequent scoring of sequence variation candidates and applying a threshold that was roughly estimated from several studies, the package reported 5 false negatives (10 % of the 51 SNPs) and 7 supposedly false positives (13 % of the 53 reported SNPs). Again, for 6 of the 46 true positive SNPs the exact position could not be determined.

Since the focus of this paper lies in generating the correct sequence variation candidates, not in evaluating the applied scoring scheme, we will not go into more detail for this part of the analysis. Still, it is remarkable that even using a rather simple scoring scheme and a rather arbitrary threshold, such sensitivity and specificity could be achieved. Note that neither scoring scheme nor applied threshold were trained for the presented example: Instead, the default scoring scheme and threshold of the analysis package were used. Currently, manual post-processing of the results is required because by manually inspecting the mass spectrometry data, the user is capable of evaluating subtle peak intensity changes that are not evaluated adequately during automated sequence variation scoring. Work on this problem is in progress.

	$k=1$	$k=2$	$k=3$
Approach presented in this paper			
Runtime for preprocessing	189 ms	292 ms	491 ms
... for constructing seq. vars.	81 ms	997 ms	29.6 s
... for scoring seq. vars.	1240 ms	30.9 s	437 s
# of constructed seq. vars.	22 447	430 114	$\approx 5 \cdot 10^6$
Runtime for scoring *one* sequence variation	55 μs	72 μs	87 μs
Total runtime	1.5 s	32.2 s	467 s
"Trivial" approach			
Approximate number of sequence variations, per base	96	8 032	308 661
Approximate number of sequence variations in total	1 132 128	$7.7 \cdot 10^7$	$2.4 \cdot 10^9$
Approximate total runtime	62.6 s	91.9 min	57 h

TABLE 1. Runtimes and numbers of sequence variations to be scored, for a genomic region of 11 793 base pairs and a test set of 12 samples.

We have depicted the runtime of the different parts of the analysis for sequence variation cost $k = 2, 3$ in Table 1. Runtime measurements were performed on a single processor desktop computer using a 1.0 GHz Pentium III processor. For $k = 1$, the trivial approach of scoring every single-base sequence variation has to score 8 sequence variations per base pair and sample. Using the argumentation of Section 3.3, we can derive approximations for the number of sequence variations that we have to score when using the "trivial" approach in case $k > 1$. This leads us to approximately $541\frac{1}{3}$ sequence variations per base pair and sample for sequence variation cost $k = 2$, and approximately $16\,676\frac{4}{9}$ sequence variations per base pair and sample for $k = 3$. For random strings, the presented numbers are slightly too high because we are dealing with finite strings (amplicons), and since some of the counted sequence variations are not minimal. On the other hand, these numbers will be much higher in most cases for biological sequences. Approximations for the runtime of scoring all potential sequence variations are also depicted in Table 1. It is obvious for $k = 1$ that the presented approach is superior to the "trivial" approach, but the latter is completely out of consideration for high throughput analysis in case $k \geq 2$ due to runtime constraints. Using the presented algorithm, though, the sequence variation analysis for $k = 3$ was performed in 0.33 seconds per analyzed mass spectrum and is therefore still suited for "real time" analysis of mass spectrometry data.

Note that the outlined scoring scheme for sequence variation candidates is comparatively simple and easy to compute. The scoring of sequence variations was implemented in C++ and highly optimized. Using more sophisticated approaches to score sequence variations will shift the runtime advantage even more into the direction of the presented approach.

6. Discussion and improvements

We have presented a computational approach to perform SNP and mutation discovery using base-specific cleavage and mass spectrometry for data acquisition. In particular, we have presented a time-efficient method to significantly reduce the number of sequence variations that have to be evaluated. Using base-specific cleavage and mass spectrometry for SNP and mutation discovery has the advantages of high throughput (4 mass spectra can be measured in 5–10 seconds) and potentially increased sensitivity and specificity

over classical Sanger Sequencing, and is in most cases sufficient to characterize and localize the detected SNPs. Potential additional advantages are the possibility to do pooling or multiplexing, see below. Better scoring schemes may enable the analysis of the cleavage mass spectra without *any* user interaction.

Besides SNP and mutation discovery, there are other applications where the approach described herein can be utilized. As an example, "re-sequencing" refers to either testing a previously sequenced region for sequencing errors, or to determine the genomic sequence of an organism (like a virus or a bacterium) when the genomic sequence of a closely related organism is known. Both applications can be tackled using the presented approach, but the sequence variation cost of interest for the latter problem might well be $k \gg 3$.

The intensity of a peak in a MS spectrum might indicate the multiplicity of the respective compomer. If we can, up to some extent, predict the multiplicity of compomers in the sample spectrum, this additional information can easily be integrated into the presented approach.

Finally, we want to point out that the presented method can easily be adopted for *pooling* as well as *multiplexing*: When pooling samples, we want to analyze *mixtures* of samples where the ratio of two present genotypes differs significantly from the 50:50 ratios expected for heterozygous samples. As an example, this is of special interest in the case of mutation discovery in potentially cancerous tissue. For multiplexing, instead of analyzing a single continuous stretch of the sample sequence of length 900 nt, we analyze, say, three distinct stretches of length 300 nt each in parallel. In both cases, only minor modifications to the presented approach are necessary.

Acknowledgments

The author wants to thank Jens Stoye and Dirk van den Boom for proofreading earlier versions of this manuscript, and Mathias Ehrich and the above for many helpful suggestions. Acquisition and manual analysis of the mass spectrometry data in Section 5 was performed by Christiane Honisch and Mathias Ehrich.

References

Apostolico, A. and Galil, Z., editors (1997). *Pattern Matching Algorithms*. Oxford University Press.

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.

Boyer, R. S. and Moore, J. S. (1977). A fast string searching algorithm. *Commun. ACM*, 20(10):762–772.

Buetow, K. H., Edmonson, M. N., and Cassidy, A. B. (1999). Reliable identification of large numbers of candidate SNPs from public EST data. *Nat. Genet.*, 21:323–325.

Cox, D. G., Boillot, C., and Canzian, F. (2001). Data mining: Efficiency of using sequence databases for polymorphism discovery. *Human Mutation*, 17:141–150.

Dausset, J., Cann, H., Cohen, D., Lathrop, M., Lalouel, J., and White, R. (1990). Program description - Centre d'Etude du Polymorphisme Humain (CEPH) - collaborative genetic-mapping of the human genome. *Genomics*, 6(3):575–577.

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. To appear in *Nucleic Acids Res.*

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

Knuth, D. E., Morris Jr., J. H., and Pratt, V. R. (1977). Fast pattern matching in strings. *SIAM J. Computing*, 6:323–350.

Mann, M., Hojrup, P., and Roepstorff, P. (1993). Use of mass spectrometric molecular weight information to identify proteins in sequence databases. *Biol. Mass Spectrom.*, 22(6):338–345.

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

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.

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

Stanssens, P., Zabeau, M., Meersseman, G., Remes, G., Gansemans, Y., Storm, N., Hartmer, R., Honisch, C., Rodi, C. P., Böcker, S., and van den Boom, D. (2003). High-throughput MALDI-TOF discovery of genomic sequence polymorphisms. Submitted.

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.

Yates III, J. (1998). Database searching using mass spectrometry data. *Electrophoresis*, 19(6):893–900.

Appendix

Lemma 2. *Given a random i.i.d. string $s = s_1 s_2 s_3 \dots$ on an alphabet of four letters. Let the random variable X denote the natural number so that the substring $s' = s_1 \dots s_X$ contains exactly three letters of the alphabet, while $s'' = s_1 \dots s_{X+1}$ contains all four letters. Then, $E(X) = 7\frac{1}{3}$.*

Proof. Counting those strings of length n that contain at most 3 out of 4 letters, we infer using inclusion/exclusion that for $n \geq 1$,

$$\begin{aligned}\mathrm{Prob}(X \geq n) =& \frac{1}{4^n}\left(\tbinom{4}{3}3^n - \tbinom{4}{2}\cdot 2^n + \tbinom{4}{1}\right)\\ =& \frac{1}{4^n}\left(4\cdot 3^n - 6\cdot 2^n + 4\right).\end{aligned}$$

The probability that the substring s' has length exactly n is

$$\begin{aligned}\mathrm{Prob}(X = n) &= \mathrm{Prob}(X \geq n) - \mathrm{Prob}(X \geq n+1)\\ &= \frac{1}{4^n}\left(3^n - 3\cdot 2^n + 3\right) \quad \text{for } n \geq 1.\end{aligned}$$

The expected length of such a substring is

$$\begin{aligned}E(X) =& \sum_{n=1}^{\infty} n\cdot \mathrm{Prob}(X = n)\\ =& \sum_{n=1}^{\infty} n\cdot\left(\frac{3}{4}\right)^n - 3\sum_{n=1}^{\infty} n\cdot\left(\frac{1}{2}\right)^n\\ &+ 3\sum_{n=1}^{\infty} n\cdot\left(\frac{1}{4}\right)^n\\ =& \frac{3/4}{(1-\frac{3}{4})^2} - 3\cdot\frac{1/2}{(1-\frac{1}{2})^2} + 3\cdot\frac{1/4}{(1-\frac{1}{4})^2}\\ =& 12 - 6 + \frac{4}{3} \;=\; 7\frac{1}{3}\end{aligned}$$

as claimed. □

SEQUENOM Inc., 3595 John Hopkins Court, San Diego, CA 92121, USA
E-mail address: boecker@CeBiTec.uni-bielefeld.de

Bisher erschienene Reports an der Technischen Fakultät
Stand: 2003-04-23

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)