BIOINFORMATICS

Vol. 23 ECCB 2006, pages e5–e12
doi:10.1093/bioinformatics/btl291

Simulating multiplexed SNP discovery rates using base-specific cleavage and mass spectrometry

Sebastian Böcker

Chair for Bioinformatics, Friedrich-Schiller-University Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany

ABSTRACT

Motivation: Single Nucleotide Polymorphisms (SNPs) are believed to contribute strongly to the genetic variability in living beings, and SNP and mutation discovery are of great interest in today's Life Sciences. A comparatively new method to discover such polymorphisms is based on base-specific cleavage, where resulting cleavage products are analyzed by mass spectrometry (MS). One particular advantage of this method is the possibility of multiplexing the biochemical reactions, i.e. examining multiple genomic regions in parallel. Simulations can help estimating the performance of a method for polymorphism discovery, and allow us to evaluate the influence of method parameters on the discovery rate, and also to investigate whether the method is well suited for a certain genomic region.

Results: We show how to efficiently conduct such simulations for polymorphism discovery using base-specific cleavage and MS. Simulating multiplexed polymorphism discovery leads us to the problem of uniformly drawing a multiplex. Given a multiset of natural numbers we want to uniformly draw a subset of fixed cardinality so that the elements sum up to some fixed total length. We show how to enumerate multiplex layouts using dynamic programming, which allows us to uniformly draw a multiplex.

Contact: boecker@minet.uni-jena.de

1 INTRODUCTION

The completion of the Human Genome Project provides researchers with a reference sequence of the human organism that covers >99% of the gene-containing regions and is highly accurate (International Human Genome Sequencing Consortium, 2004). There are several types of deviations from this reference sequence that an individual can show, among them are polymorphisms such as SNPs and mutations. SNPs (single nucleotide polymorphisms) are believed to play an important role for disease predisposition or drug side effect predisposition (International SNP Map Working Group, 2001). Mutations, observed only in certain cells or cell types of an individual, are believed to play an important role, e.g. in the development of cancer. A large fraction of today's SNP and mutation discovery is still based on Sanger *de novo* sequencing of the sample sequences of interest (Sanger *et al*., 1997; Altshuler *et al*., 2000). There exist several other techniques for polymorphism discovery, either biochemical or purely combinatorial (Buetow *et al*., 1999), each one with certain advantages and limitations.

Recently, a new method for SNP and mutation discovery was proposed, based on base-specific cleavage of DNA or RNA, and mass spectrometry (MS) to acquire the experimental data (Rodi *et al*., 2002). The experimental settings of this method have been extensively studied in literature (see for example Hartmer *et al*., 2003; Smylie *et al*., 2004). The method is commercially available (Ehrich *et al*., 2004, www.sequenom.com) and is part of a pipeline for mining disease susceptibility genes (Tang *et al*., 2004). Simulating the method's performance is essential because wet lab experiments are costly and time-consuming. Simulations allow to modify parameters such as amplicon lengths (length of the DNA region amplified by PCR) to achieve desired discovery rates, and can provide useful insight into other promising parameter modifications. Stanssens *et al*. (2004) present results of computer simulations to evaluate the method's potential using a small set of about 10 million SNP events.

In Ehrich *et al*. (2005) we consider the possibility of multiplexing the above method, i.e. polymorphism discovery in parallel for several amplicons. For example, in a three-plex we analyze three regions of length 300 nt each together, instead of a single region of length 900 nt. SNP discovery in eukaryotes is often targeted at exonic regions plus flanking UTRs (untranslated regions), that are rather short: 90% of all exons in the human NCBI database (v34.1) have length < 325 nt. Here, multiplexing can dramatically cut down reaction costs for polymorphism discovery.

Now, the question occurs whether multiplexing changes the discovery potential of the method: Can we identify the same number of polymorphisms in three 300 nt amplicons, as we can in one 900 nt amplicon? Our simulation results clearly indicate that multiplexing does not change polymorphism discovery rates (Ehrich *et al*., 2005). To compare discovery rates in a k-plex we have to select k amplicons such that the amplicons' length sum up to the desired total length. Designing primers for multiplexed PCRs is a non-trivial problem and often results in trial-and-error optimization of multiplexes. So, we ignore this step and assume that the sequences in a multiplex where randomly selected from the database. For a precise evaluation, it is important that multiplexes are drawn uniformly. Otherwise, overrepresented multiplexes can corrupt discovery rates in an unpredictable fashion.

For the analysis of measured mass spectra from base-specific cleavage experiments, we have provided computational methods for discovering sequence polymorphisms in Böcker (2003). In this paper, we show how to efficiently simulate polymorphism discovery rates of the method. We recently conducted detailed studies of the method's potential using techniques described herein where more than 275 billion SNP events were simulated (Ehrich *et al*., 2005). Furthermore, we show how to uniformly draw a multiplex with the desired conditions, to evaluate the performance of multiplexed polymorphism discovery. This problem is equivalent to drawing a subset of a multiset. Given a multiset $\mathcal{M}$ of natural numbers, a multiplex level k and a total length n we want to uniformly draw a subset of cardinality k so that the elements sum up to n. Here, we face the problem that drawing the individual elements is stochastically highly dependent. We show how to enumerate

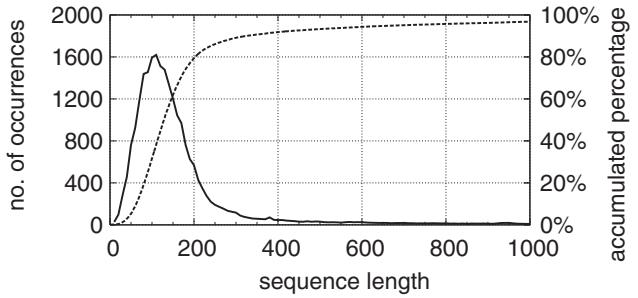

Fig. 1. Distribution of exon lengths in NCBI database (v34.1). Solid line is the number of sequences in the database (smoothed), dashed line is the accumulated percentage of sequences.

these subsets using dynamic programming, which allows us to uniformly draw such multiplexes.

2 EXPERIMENTAL SETUP AND SIMULATION OF CLEAVAGE MASS SPECTRA

Suppose we are given a target DNA molecule (or sample DNA) of length 100–2000 nt. We amplify and transcribe the sample DNA, and cleave the resulting sequence with a base-specific RNAse, such as RNAse A. After transcription in the presence of dCTP (deoxycytidine triphosphate) instead of rCTP, this endonuclease will cleave the sample sequence wherever rUTP was incorporated (Rodi *et al*., 2002). Base-specific cleavage can also be achieved using other RNAses (Hartmer *et al*., 2003), as well as biochemical methods.

We then apply MALDI (matrix-assisted laser desorption ionization) TOF (time-of-flight) MS to the products of the cleavage reaction, and extract a list of signal peaks with masses and intensities. We can repeat the above procedure, as well as the following analysis steps using cleavage reactions specific to each of the four bases, and we obtain up to four mass spectra, each corresponding to a base-specific cleavage reaction (Fig. 1).

Often, experimentalists will not use four distinct cleavage reactions to obtain cleavage patterns for all bases, but instead cleave two bases on the forward strand and the same two bases on the reverse strand. For example, we cleave base C or base T, both on the forward strand and on the reverse strand, as shown in Figure 3. Such cleavage can be achieved using RNAse A (Rodi *et al*., 2002). The analysis and simulation of such cleavage is mainly identical to analyzing cleavage reactions on the forward strand only, so we omit the details for the sake of brevity.

Simulating the mass spectrum that results from a base-specific cleavage experiment is relatively simple and can be compared with simulating the mass spectrum of a trypsin-digested protein. To this end, given a sample sequence s we sum up the masses of characters until we reach cleavage character x. Then, we add the resulting fragment mass to a list, and continue generating the next fragment mass. Finally, we sort this peak list with respect to mass, joining fragments of identical mass by generating a fragment with higher intensity. We also take into account mass modifications resulting from the utilized cleavage biochemistry. Unlike Peptide Mass Fingerprint spectra of digested proteins, DNA/RNA mass spectra from base-specific cleavage show very good agreement with those predicted *in silico*, for example as shown in Figures 2 and 4 of Ehrich *et al*. (2005). We can therefore safely ignore additional and missing peaks in our simulations. A comparable mass spectra prediction has been successfully used for the analysis of measured SNP discovery mass spectra (Ehrich *et al*., 2004).

We can easily modify this procedure for partial cleavage, where the cleavage biochemistry is modified in a way such that not all cut bases but only a certain percentage will be cleaved, see Böcker (2004) for details. In this case, we also generate fragments that contain up to K uncleaved characters x, for some fixed threshold K. We can do so in time $O(K|S|)$ and since the output size (the number of fragments) is of the same order, this simple approach is optimal. In applications, runtime can be slightly decreased by generating masses of complete cleavage fragments first, then using these for computing partial cleavage fragment masses. Doing so, we achieve a runtime of $O(m + |S|)$ where m is the size of the output peak list before merging peaks.

Next, we have to take into account the mass range of the mass spectrometer by discarding all peaks outside this mass range. We may also want to include low intensity peaks in our simulation. Random elongation during transcription adds one additional base to the transcript and is responsible for up to four additional peaks. Peaks can also be due to abortive cycling where the transcriptase 'falls off' after 1–20 bases at the beginning of the sequence, but this effect is usually countered by reasonable primer design.

We have depicted a measured mass spectrum, together with masses of simulated cleavage products in Figure 2. As one can see, there is a high agreement between predicted and measured mass spectrum, while measured peak intensities vary. Hence, only the presence or absence of a peak is usually regarded as a save indication of a sequence polymorphism.

3 MASS SPECTRA CHANGES FROM SEQUENCE POLYMORPHISMS

Let s' be a sequence with edit distance one from our original sample sequence s. Now, we can compute the mass spectra of s' for all cleavage reactions, and compare those with the original spectra of s. In the following, we indicate a faster way to compare the mass spectra.

To this end, consider a certain cleavage reaction with cleavage character x. Comparing sequences s and s', there is a position j of s such that character σ in s is replaced by $\sigma' \neq \sigma$ in s' for $\sigma, \sigma' \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}, \epsilon\}$, the latter corresponding to insertions and deletions. Let i, i' denote the largest (smallest) position before (after) position j of s where the cleavage character x can be found. Now, it is sufficient to simulate the mass spectrum from position i to i' of s, with and without the sequence polymorphism. We discard those peaks with masses outside the mass range. For complete cleavage, if $x \neq \sigma, \sigma'$ then the polymorphism changes the mass of a single fragment; if $x = \sigma$ then two fragments are merged by the polymorphism; and if $x = \sigma'$ then one fragment is divided into two. For partial cleavage, at most $3K + 3$ fragments are affected by the polymorphism.

We repeat the above simulations for all cleavage reactions. Ultimately, we join all mass spectra changes, keeping track of the cleavage reaction that every peak stems from. In the following, we assume that 'reaction' is simply a cardinal number. We call the resulting list of peaks (with mass, intensity change and reaction) the fingerprint of the polymorphism, as shown in Figure 3.

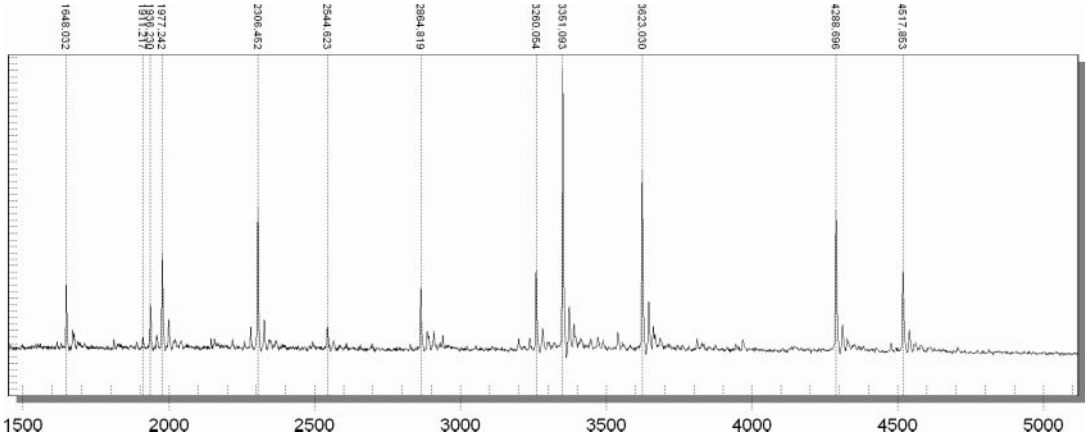

Fig. 2. Simulated and measured mass spectra. Peaks in simulated spectrum are indicated by dotted lines. Except for the low mass region, spectra show a high agreement.

```
forward strand  5'...CAATAGGC[A/G]TTAGGCCA...3'
reverse strand  3'...GTTATCCG[T/C]AATCCGGT...5'
```

Cleavage:	A–allele	G–allele
C forward	ATTAGGC	GTTAGGC
T forward	AGGCAT	AGGCGT
C reverse	GTAATC	GC AATC
T reverse	CCGT AAT	CCGCAAT

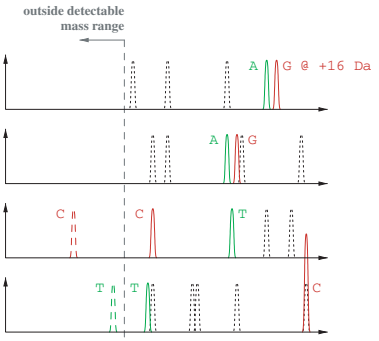

Fig. 3. Fingerprint of a polymorphism. RNAse A cleavage on forward and reverse strands for an A/G polymorphism on the forward strand. Differing cleavage fragments and resulting peaks are displayed; in a heterozygous sample, all peaks are present simultaneously with reduced intensities.

All peaks in a fingerprint would lead to the detection of the polymorphism, in case peak intensities can be accurately predicted. Unfortunately, the prediction of peak intensities in DNA MS is a non-trivial problem as it is for peptide MS (Elias *et al*., 2004). So, we usually regard only those peaks of a fingerprint as indications of a polymorphism, where new peaks are generated when compared with the reference sequence spectra, or peaks from these spectra are missing. For example, in Figure 3 the rightmost peak in the fourth reaction only changes in intensity. In addition, we face the problem of silenced peaks, due to mass inaccuracies and resolution constraints. For ideal MS, we are able to distinguish between peaks with arbitrarily small mass difference. In reality, we can measure masses with some mass accuracy and if two peaks in a mass spectrum get too close (∼2–10 Da for linear TOF MS), they merge into a joined peak with higher intensity and mass in-between the masses of the original peaks. So, a peak with mass m in the peak pattern can be impossible to detect because of some peak with mass $m' = m + \delta$ in the reference mass spectrum. To exclude silenced peaks we have to scan through the original peak list of size $O(|S|)$.

For homozygous samples, only the polymorphism sequence s' is present in the sample. But for heterozygous samples, both s' and the original sequence s are present, so all peaks of s will be present in the mass spectrum. In this case, only additional peaks indicate a polymorphism, reducing the number of polymorphisms that can be discovered.

For a fixed reaction and complete cleavage, we can compute a fingerprint by identifying the fragments affected by a polymorphism, then computing the masses of fragments if the polymorphism is present. The time-consuming tasks in doing so are, first, identifying the affected fragments and, second, dividing a fragment into two, in case a new cleavage character results from the polymorphism. But if we assume that we iteratively simulate polymorphisms for all positions, then both of these tasks can be accomplished in constant time, by updating divided fragment masses and eventually proceeding with the next affected fragment. For partial cleavage and R cleavage reactions we compute the $O(K \cdot R)$ peaks of the fingerprint in $O(K \cdot R)$ time. Identifying silenced peaks using binary search leads to an overall time complexity of $O(K \cdot R \log |S|)$.

4 RUNTIME HEURISTIC FOR UNIQUE MASS FINGERPRINTS

In addition to the question whether a polymorphism can be discovered, we usually also want to know whether we can uniquely identify it: That is, is there another polymorphism with identical or indistinguishable fingerprint? In this case, we can detect both polymorphisms but we cannot tell them apart. Assume that we have already computed and stored the fingerprints of all N polymorphisms under consideration. Using pairwise comparisons we can find unique fingerprints in time quadratic in N.

To speed up this part of the analysis, an obvious way to go is to sort the fingerprints with respect to some order, then to search for identical neighbors in the sorted list. This would improve the runtime to $O(N \log N)$ but leaves us with the following problem: When comparing two fingerprints $\mathcal{F}$ and $\mathcal{F}'$, we again have to take into account mass inaccuracies in the MS read. So, we say that a fingerprint $\mathcal{F}'$ covers a fingerprint $\mathcal{F}$ if for every peak in $\mathcal{F}$ with mass m, there exists a peak in $\mathcal{F}'$ with identical reaction and mass m' such that $|M - M| < \delta^*$ holds. Two fingerprints $\mathcal{F}, \mathcal{F}'$ are indistinguishable if $\mathcal{F}$ covers $\mathcal{F}'$, and $\mathcal{F}'$ covers $\mathcal{F}$. Hence, we are counting those fingerprints that are distinguishable from any other fingerprint.

We proceed as follows: We first collect all peak masses from all fingerprints under consideration, and sort them regarding reaction and mass. Next, we construct a map from these mass/reaction tuples to indices, so that two peaks with same reaction and mass difference below δ also receive the same index. Still, two masses with identical index may differ by more than δ even for identical reaction. For one reaction, the three masses 1000, 1001 and 1002 with $\delta = 1.5$ are mapped to the same index even though $1002 - 1000 = 2 > \delta$. Now, we can map every fingerprint to the corresponding index fingerprint, and the above implies that indistinguishable fingerprints also have identical index fingerprints, while the converse is not necessarily true.

We sort the fingerprints according to their index fingerprints, and in the following step we check whether any two fingerprints with identical index fingerprint are truly indistinguishable. The worst-case runtime of this approach is unfortunately still quadratic, because all fingerprint may show identical index fingerprints and we are left with the pairwise comparisons of fingerprints mentioned above. But for actual simulations using biological data the fingerprinting technique efficiently speeds up runtimes.

5 MULTIPLEXING AND LAYOUTS

Multiplexed polymorphism discovery using base-specific cleavage and MS (Ehrich *et al*., 2005) allows to analyze several amplicons in parallel, thereby reducing the cost per base of the experiments. The experimental setup from Section 2 is modified such that not a single amplicon, but two or more amplicons are amplified together, by providing $2k$ PCR primers for the k amplicons. This mixture is then cleaved and analyzed by MS, as in the case of a single-plex reaction. Estimating the discovery rate for a multiplex can be done just as in the case of a single amplicon, merging the peak lists of the individual amplicons, for every reaction.

Now, the experimental setup requires primer design for a multiplexed PCR, where certain sets of amplicons are amplified together. We may want to choose these amplicon multiplexes in a way that maximizes discovery rates, but face the following problem: although primer design for multiplexed PCRs is used broadly for, say, high-throughput genotyping (Sharan *et al*., 2005), it still remains a non-trivial problem (Chamberlain and Chamberlain, 1994; Edwards and Gibbs, 1994). In particular, multiplex primers designed *in silico* have to be evaluated experimentally. Elnifro *et al*. (2000) state that empirical testing and a trial-and-error approach may have to be used, because there are no means to predict the performance characteristics of a primer pair. We refer the reader to the literature for more details on multiplex primer design issues.

In the absence of efficient computational methods for reliable multiplex primer design, we assume that amplicon multiplexes are not selected in a way that maximizes discovery performance but instead, in a way that solely assures reliable multiplexed PCR amplification. Reliable amplification and discovery performance are presumably uncorrelated processes, so we assume in our simulations that every such multiplex is uniformly drawn from the set of all multiplexes that have the required overall length. Note that all multiplexes must be drawn with equal probability, because overrepresented multiplexes may corrupt discovery rates in an unpredictable fashion. Randomly drawing multiplexes can be seen as a worst-case scenario for discovery rates because we deliberately ignore (trial-and-error) optimization potential.

We now describe how to uniformly draw from the set of multiplexes. Formally, the problem is as follows: We are given a set S of sequences and a multiplexing order k. We want to uniformly draw a subset of k sequences such that the summed lengths of these sequences equal the fixed total length. We cannot iteratively draw the sequences because these drawings are highly dependent; in particular, for the last sequence the previous sequences pinpoint the exact sequence length. It should be understood that to uniformly draw such multiplex of sequences we do not have to know the actual sequences in S but instead, it suffices to know just the sequence lengths.

To this end, we transform S into a multiset of natural numbers: A *multiset* $\mathcal{M} = (M, n)$ is formally represented by the set M together with the multiplicity map $n : M \to \mathbb{N}$. For example, the multiset $\mathcal{M} = \{1, 2, 2, 3\}$ is represented by $M = \{1, 2, 3\}$ and $n(1) = 1$, $n(2) = 2$ and $n(3) = 3$. We denote the sum of elements in a multiset $\mathcal{M} = (M, n)$ by $\sum \mathcal{M} := \sum_{m \in \mathcal{M}} m = \sum_{m \in M} n(m) \cdot m$. A multiplex or k-plex of $\mathcal{M}$ is simply a sub-multiset $\mathcal{M}' \subseteq \mathcal{M}$ of cardinality $\mathcal{M}'k$. For a fixed total length L, the set of multiplexes to draw from is

$$mp(\mathcal{M}, k, L) = \left\{ \mathcal{M}' \subseteq \mathcal{M} \,:\, \mathcal{M}'k,\ \sum \mathcal{M}' = L \right\}.$$

Now, the task of drawing a multiplex of sequences can be split into two steps. First, we uniformly draw a multiplex $\mathcal{M}' \subseteq \mathcal{M}$ with the desired properties from the above set. Second, we transform the multiplex $\mathcal{M}' = (\mathcal{M}', n')$ into a multiplex of sequences, by drawing $n'(l)$ sequences of length l for every $l \in M'$. To do so, we preprocess the sequences in S by sorting them into buckets with respect to their length. Then, drawing $n'(l)$ sequences of length l can be done in time $O(n'(l))$ using a hash table. In total, the

$$k = \mathcal{M}' \text{ sequence scan bedrawn in time } O(k).$$

So, the problem we are left with is to uniformly draw a multiplex from $\mathcal{M}$ respecting the multiplicities of elements in $\mathcal{M}$. To simplify notations, we introduce the notion of layouts: a layout $x = (x_1, \ldots, x_k) \in \mathbb{N}^k$ is a k-tuple of natural numbers satisfying $x_1 \leq x_2 \leq \cdots \leq x_k$. Let $x(l)$ denote the number of indices

$j = 1, \ldots, k$ such that $x_j = l$. Every multiset $\mathcal{M}$ of natural numbers can be mapped to a unique layout by sorting the numbers in $\mathcal{M}$. Clearly, every layout also corresponds to a unique multiset, so we concentrate on layouts instead of multiplexes in the following.

Finally, we introduce the cardinality of multiplexes and layouts. Reconsider the set of sequences S we started from, and let $\mathcal{M}' = (M', n')$ be a multiplex of the corresponding multiset $\mathcal{M}$. How many sequence multiplexes $S' \subseteq S$ correspond to this multiplex $\mathcal{M}'$? For every length $l \in M'$ there exist $n(l)$ sequences to draw from, so we can choose $n'(l)$ sequences in $\binom{n(l)}{n'(l)}$ ways. Multiplying over all $l \in M'$ shows that there exist $\mathrm{card}(\mathcal{M}') = \prod_{l \in M'} \binom{n(l)}{n'(l)}$ sequence multiplexes corresponding to the multiplex $\mathcal{M}'$. Note that $\mathrm{card}(\mathcal{M}') > 0$ if and only if $n'(l) \leq n(l)$ holds for all $l \in M'$. Analogously, we define the cardinality of a layout $x = (x_1, \ldots, x_k)$ by

$$\mathrm{card}(x) := \prod_{l \in \{x_1, \ldots, x_k\}} \binom{n(l)}{x(l)}. \tag{1}$$

The simplest way to uniformly draw a multiplex, is to lexicographically order the layouts, and for every layout x store the summed cardinalities of all layouts smaller or equal to x. Then, uniformly drawing a layout can be achieved by drawing a random number, and looking it up in a sorted table. Unfortunately, there exist up to $\binom{k + \max \mathcal{M} - 1}{k}$ layouts of size k, so we do not want to store all layouts in memory, see Section 7. Clearly, the number of layouts also prohibits computation of layout cardinalities on-the-fly to draw a layout.

6 ENUMERATING MULTIPLEXES

Let $\mathcal{M} = (M, n)$ be the multiset we want to draw from, and let $l*$ denote the largest integer in $\mathcal{M}$. For the sake of readability, we ignore the 'total length constraint' for the moment, and concentrate solely on computing cardinalities of layouts. To this end, let $D[l, j]$ be the summed cardinality of all layouts $(x_1, \ldots, x_j)$ of size j that start with $x_1 = l$. Let $E[l, j]$ be the summed cardinality of all layouts of size j that start with $l' \geq l$, so $E[l, j] = D[l, j] + E[l + 1, j]$. The main finding of this section is that D, E can be easily computed using the recurrence relation

$$D[l,j] = \sum_{i=1}^{j} \binom{n(l)}{i} E\big[l+1, j-i\big] \tag{2}$$

for $l = l^*, \ldots, 2, 1$. Initial values are $E[l, 0] = 1$ for all $l = 1, \ldots, l^*, + 1$, and $E[l^*, + 1, j] = 0$ for all $j \geq 1$. Note that $E[l, j] = D[l, j] + E[l + 1, j]$ for $l \leq l^*$, can be computed in constant time.

The proof of recurrence (2) is based on the observation that for any layout x with $x_1 = l$, there exists an integer $1 \leq k' \leq k$ such that $x_j = l$ for $j = 1, \ldots, k'$ and $x_j > l$ for all $j > k'$. So, the layout can be constructed by concatenating k' times l plus a layout of size $k - k'$ that starts with $l' > l$. See the Appendix for an example. We can compute the tables D, E in time $O(k^2\ l^*)$, as every step of the summation in (2) takes only constant time, since $\binom{n(l)}{1} = n(l)$ and $\binom{n(l)}{i+1} = \frac{n(l) - i}{i+1} \binom{n(l)}{i}$.

We uniformly draw a layout using tables D, E as follows: Choose a random integer r between 1 and $E[1, k]$. Find the largest l such that $r \leq E[l, k]$ holds. Now, $x_1 = l$ is the initial element of our layout, but we have to find its multiplicity: Set $r \leftarrow r - E[l + 1, k]$. We proceed along recurrence relation (2): For $i = 1, \ldots, k$, if $r \leq \binom{n(l)}{i} E\big[l+1, k-i\big]$ then break the loop; otherwise, set $r \leftarrow r - \binom{n(l)}{i} E\big[l+1, k-i\big]$ and continue. Now, our layout starts $x_1 = x_2 = \cdots = x_{k-i} = l$, and we have to find the remaining layout $x_{k-i+1}, \ldots, x_k$. This can be done iteratively, by searching for a layout of size i starting with $l' > l$. Finally, we output the concatenated layout. The complete process of drawing a layout can be done in time $O(k \log l^*)$ or $O(k + l^*)$, using binary search in E or linear search in D, respectively.

Without length constraint, we can easily draw a sequence k-plex by iteratively drawing k sequences from S, so there is no need to use tables D, E for doing so. But we will see in the next section that the above algorithm can be generalized to drawing multiplexes with length constraint.

Now, we want to enumerate multiplexes with length constraints. Let $\mathcal{M} = (M, n)$ be the multiset we want to draw from, and let l^* denote the largest integer in $\mathcal{M}$. Now, we want to take into account the length constraint and only draw multiplexes that sum up to L^*. Let $F[l, j, L]$ be the summed cardinality of all layouts of size j that start with l and sum up to L: $F[l,j,L] = \sum_{x \in X} \mathrm{card}(x)$ where X is the set of layouts $(x_1, \ldots, x_j)$ satisfying $x_1 = l$ and $\sum_i x_i = L$.

Analogously to above, let $G[l, j, L]$ be the summed cardinality of all layouts of size j that start with $l' \geq l$ and sum up to L. These matrices F, G can be computed using the recurrence relation

$$\begin{aligned} F\big[l,j,L\big] &= \sum_{i=1}^{min\{j,\, L/l\}} \binom{n(l)}{i} G\big[l+1, j-i, L-i\cdot l\big] \\ G\big[l,j,L\big] &= F\big[l,j,L\big] + G\big[l+1,j,L\big] \end{aligned} \tag{3}$$

for $L = 0, \ldots, L^*, j = 1, \ldots, k$ and $l = l^*, \ldots, 2,1$. Initial values for matrix G are $G[l, 0, L] = 1$ if $L = 0$ and $G[l, 0, L] = 0$ if $L > 0$, as well as $G[l^* + 1, j, L] = 0$ for $j \geq 1$ and all L. The proof of recurrence (3) is analogous to above. We can compute the tables F, G in time $O(k^2 l^* L^*)$ and need $O(k l^* L^*)$ space to store them. We uniformly draw a layout using these tables, taking into account the sum of layout entries, in time $O(k \log l^*)$ or $O(k + l^*)$, respectively.

For multiplexed polymorphism discovery, we must discard multiplexes $\mathcal{M}'$ where $\min \mathcal{M}'$ and $\max \mathcal{M}'$ diverge too much, because such amplicons cannot be amplified in a single PCR. We omit the details.

7 COMPUTATIONAL RESULTS

To evaluate the polymorphism discovery performance of the method, we randomly selected sequences of the desired length from the 4 Mb genetic region around ApoE (Lai *et al.*, 1998). Then, we simulated four cleavage reactions for every potential sequence variation, and checked whether the polymorphism can be detected under a certain parameter setting. As indicated above, both wild-type and polymorphism signals are present for heterozygous polymorphisms, so only additional signals are used to identify heterozygous polymorphisms and, hence, discovery rates are better in the case of homozygous polymorphisms.

In Figure 4, we show simulation results for three different parameter settings. In the upper two figures, solid triangles indicate

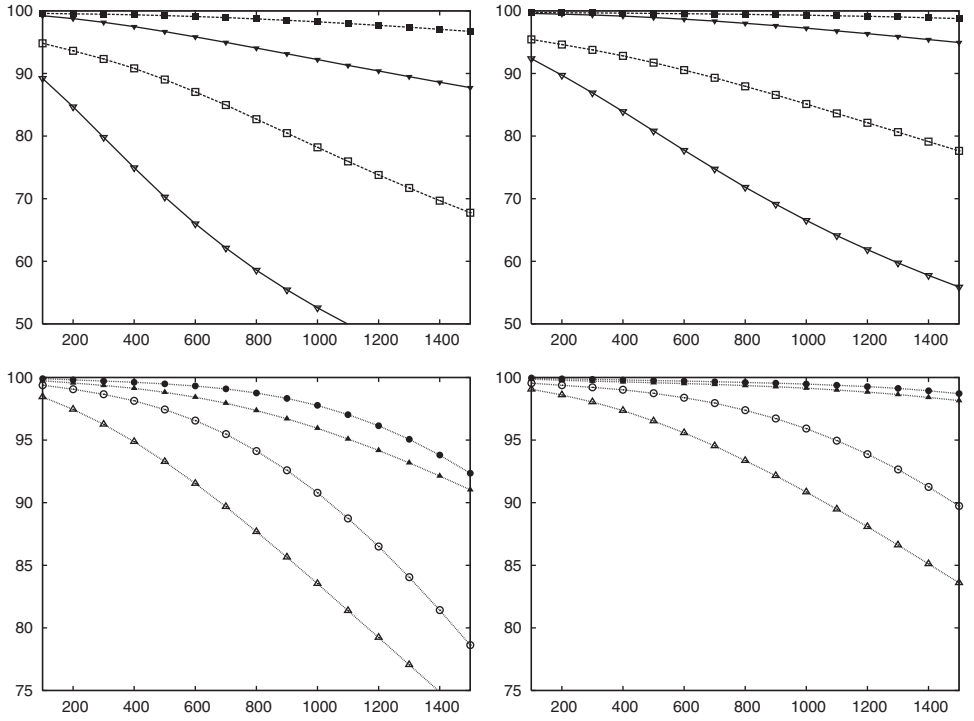

Fig. 4. Simulation results. Percentage of polymorphisms (y-axis) that can be discovered (solid points) and uniquely identified (outlined points) when sequence lengths vary (x-axis). Heterozygous polymorphisms (left) and homozygous polymorphisms (right). Observe the different scalings of the y-axis in the figures. See text for details on simulation settings.

'conservative' parameter settings for complete cleavage, where every peak silences the detection of other peaks in a broad area of several Dalton, and intensity changes are not used for polymorphism discovery. Here, we assume a polymorphism to be detectable if there is at least one additional peak (one additional or missing peak, in the case of heterozygous samples on the right) present in at least one of the mass spectra. That is, the peak is inside the admissible mass range, and is not silenced by another peak. Solid boxes indicate simulation parameter settings where, in addition, intensity changes of >50% are accepted for polymorphism detection. In the lower two figures, solid triangles indicate 'conservative' parameter settings for partial cleavage, where we assume that only partial cleavage fragments of first order $K = 1$ can be detected by the mass spectrometer. Finally, solid circles represent partial cleavage with fragment order $K = 2$. Outlined triangles, boxes and circles indicate the percentage of polymorphisms that can be uniquely identified from their mass fingerprint, see Section 4. Figures on the left show results for heterozygous polymorphisms, figures on the right homozygous polymorphisms. A more detailed description of simulation parameters, together with additional simulation results are in preparation.

Regarding the identification of unique fingerprints, the indexing technique of Section 4 resulted in an overall speedup of simulation runtimes by 65–80%, i.e. the runtime improved by a factor of 3-fold to more than 5-fold (data not shown). With this optimization, the total runtime for simulating a single SNP was ∼60 μs on a UltraSparc III processor with 750 MHz, resulting in a runtime of 0.5 s for an amplicon of length 1000.

Table 1. Memory requirements for drawing multiplexes of length $L = 1500$

Multiplex level	$k = 3$	$k = 5$	$k = 7$	$k = 9$
Amplicon lengths	354–707	213–424	152–303	118–235
No. of sequences	12 601	48 452	1 15 870	1 47 113
Naive approach	1.4×10^5	8.6×10^6	1.1×10^9	4.4×10^{10}
DP approach	1.6×10^6	1.6×10^6	1.5×10^6	1.6×10^6

Number of entries that have to be stored in memory, for the naive approach (layouts) and the dynamic programming approach (size of table). Exonic sequences from NCBI database (v34.1) including 40 nt UTR.

For the multiplexing study conducted in Ehrich *et al*. (2005), we used all exons from the human NCBI database (v34.1) with 20 nt flanking UTR regions on both sides. For a multiplex of total length L, we discarded all sequences with length outside the interval from $(1/\sqrt{2})L/k$ to $\sqrt{2}L/k$. Our simulations show that there is no significant difference in discovery rates between single-plexed and multiplexed polymorphism discovery, see Ehrich *et al*. (2005) for details.

For comparison, we also implemented the naïve approach of enumerating all layouts with summed cardinality. In both approaches, the runtime for preprocessing as well as drawing a multiplex is negligible compared with the MS simulations, so we omit the details. On the contrary, memory requirements of the approaches differ strongly, as expected, refer the details given in Table 1. In fact, we did not draw multiplexes with exact total

length constraint L but instead used an admissible window of $\sim$20 nt. In this case, memory requirements of the naïve approach grow linear with the window size, but do not change for our dynamic programming approach.

8 CONCLUSION

We presented a method for simulating polymorphism discovery rates for base-specific cleavage and MS. Such simulations are of great interest to experimentalists in order to evaluate the potential of this method for polymorphism discovery, to find promising parameter modification and, finally, to estimate discovery rates for a particular polymorphism discovery experiment. In particular, simulations can help to maximize discovery rates for multiplexed experiments, where multiplexed PCR primer design allows for several ways of multiplexing sequences together.We have also introduced an algorithm for uniformly sampling multiplexes with length constraints. Our sampling method reduces space requirements to a point where its application is possible on any personal computer. In addition, recurrences (2) and (3) can be adopted for counting multiplexes when even more complex constraints must be satisfied. We hope that the presented method is of use for applications beyond the simulation of polymorphism discovery rates.

ACKNOWLEDGEMENTS

The author was supported by 'Deutsche Forschungsgemeinschaft' (BO 1910/1-1) within the Computer Science Action Program. Additional programming was done by Matthias Steinrücken.

REFERENCES

Altshuler,D. *et al*. (2000) An SNP map of the human genome generated by reduced representation shotgun sequencing. *Nature*, **407**, 513–516.

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

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

Buetow,K.H. *et al*. (1999) Reliable identification of large numbers of candidate SNPs from public EST data. *Nat. Genet.*, **21**, 323–325.

Chamberlain,J.S. and Chamberlain,J.R. (1994) Optimization of multiplex PCRs. In Mullis,K.B., Ferre,F. and Gibbs,R.A. (eds), *The Polymerase Chain Reaction*. Birkhauser, Boston, MA, pp. 38–46.

Edwards,M.C. and Gibbs,R.A. (1994) Multiplex PCR: advantages, development, and applications. *PCR Methods Appl.*, **3**, S65–S75.

Ehrich,M. *et al*. (2004) SNP discovery using the MassARRAY system. SEQUENOM Application Note.

Ehrich,M. *et al*. (2005) Multiplexed discovery of sequence polymorphisms using base-specific cleavage and MALDI-TOF MS. *Nucleic Acids Res.*, **33**, e38.

Elias,J.E. *et al*. (2004) Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nat. Biotechnol.*, **22**, 214–219.

Elnifro,E.M. *et al*. (2000) Multiplex PCR: optimization and application in diagnostic virology. *Clin. Microbiol. Rev.*, **13**, 559–570.

Hartmer,R. *et al*. (2003) RNAse T1 mediated base-specific cleavage and MALDI-TOF MS for high-throughput comparative sequence analysis. *Nucleic Acids Res.*, **31**, e47.

International Human Genome Sequencing Consortium. (2004) Finishing the euchromatic sequence of the human genome. *Nature*, **431**, 931–945.

International SNP Map Working Group. (2001) A map of human genome sequence variation containing 1.4 million SNPs. *Nature*, **409**, 928–933.

Lai,E. *et al*. (1998) A 4-Mb high-density single nucleotide polymorphism-based map around human ApoE. *Genomics*, **54**, 31–38.

Rodi,C.P. *et al*. (2002) A strategy for the rapid discovery of disease markers using the MassARRAY system. *BioTechniques*, **32**, S62–S69.

Sanger,F. *et al*. (1977) DNA sequencing with chain-terminating inhibitors. *Proc. Natl Acad. Sci. USA*, **74**, 5463–5467.

Sharan,R. *et al*. (2005) Multiplexing schemes for generic SNP genotyping assays. *J. Comput. Biol.*, **12**, 514–533.

Smylie,K.J. *et al*. (2004) Analysis of sequence variations in several human genes using phosphoramidite bond DNA fragmentation and chip-based MALDI-TOF. *Genome Res.*, **14**, 134–141.

Stanssens,P. *et al*. (2004) High-throughput MALDI-TOF discovery of genomic sequence polymorphisms. *Genome Res.*, **14**, 126–133.

Tang,K. *et al*. (2004) Mining disease susceptibility genes through SNP analyses and expression profiling using MALDI-TOF mass spectrometry. *J. Proteome Res.*, **3**, 218–227.

APPENDIX

Example of the layouts of a multiset

Consider the multiset $\mathcal{M} = \{1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3\}$ with $n(1) = 3$, $n(2) = 4$ and $n(3) = 5$. In this case, there exist 10 layouts and a total 220 three-plexes, as listed in the following table:

layout x	3,3,3	2,3,3	2,2,3	2,2,2	1,3,3
card (x)	10	40	30	4	30

layout x	1,2,3	1,2,2	1,1,3	1,1,2	1,1,1
card (x)	60	18	15	12	1

Hence, there exist 10 three-plexes with layout (3, ·, ·); 40 + 30 + 4 = 74 three-plexes with layout (2, ·, ·) and $30 + L + 1 = 136$ three-plexes with layout (1, ·, ·).

For three-plexes, the recurrence tables D, E for this example are as follows (initial values in gray):

$D[l, j]$	$j = 1$	2	3
$l = 3$	5	10	10
2	4	26	74
1	3	30	136

$E[l, j]$	$j = 0$	1	2	3
$l = 4$	1	0	0	0
3	1	5	10	10
1	1	12	66	220

Compare column $D[\cdot, 3]$ with our above calculations.

Fixing a total multiplex length $L^* = 7$ only layouts (2, 2, 3) and (1, 3, 3) have the correct sum, so $F[3, 3, 7] = 0$, $F[2, 3, 7] = 30$ and $F[1, 3, 7] = 30$ holds.