Deconvolving Sequence Variation in Mixed DNA Populations

Andy Wildenberg Steven Skiena* Pavel Sumazin
State University of New York at Stony Brook
Stony Brook, NY 11794-4400 USA
{awilden,skiena,psumazin}@cs.sunysb.edu

1. INTRODUCTION

The need for DNA sequencing did not end with the successful public and private projects to sequence the human genome. Indeed, attention is shifting from de novo sequencing of new organisms to analyzing sequence variation for research and diagnostic purposes.

Contemporary electrophorisis-based sequencing machines produce curves registering the amount of each of the four nucleotide bases as a function of sequence position (see Figure 1). For homogeneous DNA samples, the largest peaks at each position define the underlying sequence. However, more careful analysis of sequence trace data holds promise for determining the presence and frequency of mutations in inhomogeneous samples.

In this paper, we look at the problem of using sequence trace data to identify sequence variants in mixed DNA populations. Our work is motivated by a new line of capillary electrophorisis sequencing machines being developed by BioPhotonics Corporation [10]. By using advanced single-photon detectors and other technologies, BioPhotonics has the capability to not only detect but accurately determine the relative frequency of each base at each position to within 10%, and expects to reduce this error rate to 1% in the near future.

This motivates a variety of questions concerning how accurately we can sequence mixed populations from a single sample using relative frequency information. Possible applications of this technology include:

- *Frequency of Acquired Mutations* – Perhaps the greatest promise of modern genomics is that of individualized medicine, where an individual's genetic composition is determined and analyzed to determine the best course of treatment. New technologies such as microarrays [5, 8] offer promise for obtaining sequence and expression data on an individual scale. Microarray studies of leukemia and breast cancer [1, 9] tissues have demonstrated that cancer subtypes can be accurately diagnosed on the basis of genomic data, and with them the prognosis for survival under various treatments.

  Such microarray studies will continue to help develop our understanding of gene expression and disease. However, the technologies used for widespread diagnostic tests may well be different, to minimize costs and increase robustness. Indeed, a major goal of BioPhotonics efforts is developing smaller, cheaper DNA sequencing machines with the vision of placing them in doctor's offices for diagnostic applications.

  Particularly important for many medical applications is the need to analyze sequence from heterogeneous genomic samples. Such mixed populations naturally arise from acquired mutations, say, in cancer, where various mutations to oncogenes such as p53 can lead to dramatically different disease courses. Extensive databases of p53 mutations are being constructed, including [2, 13, 14].

  In this paper, we provide simulation results demonstrating our ability to identify p53 mutations as a function of mutation frequency and sequencing accuracy.

- *SNP Generation and Analysis* – Single-nucleotide polymorphisms (SNPs) represent an important part of sequence variation in humans. Cataloging SNPs is an important problem in contemporary sequence analysis [11]. Here we propose a potentially high-throughput technology to catalog SNPs. A pool of DNA from m distinct individuals is assembled, with a region of interest amplified using PCR. Sequencing the resulting product and deconvolving the results will be significantly more efficient than individual sequencing runs for large m, provided they can be accurately analyzed for large m.

*Corresponding author. This work is partially supported by NSF Grant CCR-9988112 and ONR Award N00149710589.

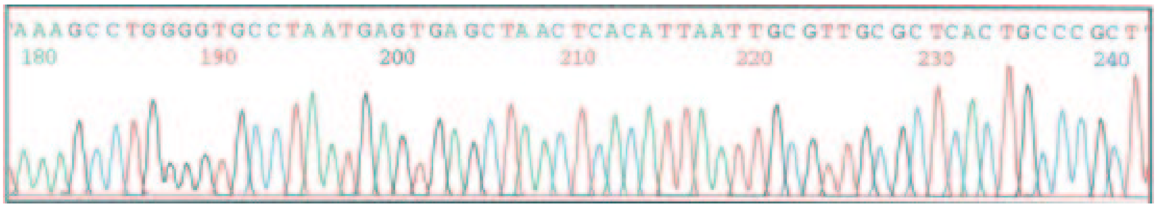

Figure 1: Trace data from a DNA sequencing machine

In this paper, we study the potential of this approach both theoretically and through simulation. We demonstrate that, under reasonable assumptions of polymorphism rates and error probabilities, pool sizes of over 100 people can be analyzed on a single sequencing run.

- *Viral Population Analysis* – Viruses such as HIV evolve rapidly, and each infected patient soon comes to host a variety of different strains. Our techniques make it possible to determine the various mutations present in a sample, as well as their relative frequencies. Obtaining accurate viral population frequency data will be important to determine a patient's response to a given course of treatment, and determine which strains react best to a given therapy.

  In this paper, we demonstrate that accurate determination of the relative frequencies of four distinct strains can be made even in the face of base-frequency error rates up to 25%.

1.1 Problem Definitions

If the population is comprised of two sequences which differ only in substituting one base position then identifying the single nucleotide polymorphism (or SNP) is in principle easy: just look for the single base position with two peaks. But how can we deconvolve more complicated mutation populations, with insertion and deletion operations? In this paper, we look at the algorithmic complexity of several sequence deconvolution problems – giving efficient algorithms where they exist and hardness proofs where they do not.

The act of deconvolving mutations from sequence trace data can be partitioned into three distinct problems:

- *Base calling* – The problem of calling bases from electrophorisis traces is complicated by a variety of technology-dependent factors. Programs such as Phred [6], TraceTuner [4], and LifeTrace [16] successfully analyze trace data, and assign each base an associated quality level or error probability.

  In this paper, we assume the trace analysis is performed by an external program, which returns the percentage of each base observed at each position for subsequent analysis. We will assume that such data is subject to errors, as discussed in Section 4.

- *Mutation deconvolution* – Given the set of observed bases at each position, which mutations are present in the given sample? In medical applications we can assume that the general wildtype sequence is known, as well as a catalog of commonly occurring or previously-studied mutations. Formally, our problem is:

  Input: A wildtype sequence S, a set V of allowable variations on S, and an experimental profile, consisting of an ordered sequence of n subsets on alphabet Σ.

  Output: The smallest subset $V' \subset V$ such that the character position subsets of V' and S yield exactly the input profile.

  Consider the following example, where wildtype string $S = ACTGTTGACTCATCC$ gives rise to the following profile:

```
S:  ACTGTTGACTCATCC
     AGTC  CTCATCG C
```

  One solution with three mutations explaining this profile consists of a 4-base substitution starting at position 2 with the sequence AGTC, a deletion of a A at position 8, and an insertion of a G at position 14.

```
S: ACTGTTGACTCATCC    wildtype
   aAGTCtgactcatcc    Sub(2,AGTC)
   actgttgCTCATCc     Del(8,1)
   actgttgactcatGcC   Ins(14,G)
```

  Bases in the mutations that contribute to the profile are capitalized, and the portion of the mutation that has changed is underlined.

- *Population frequency determination* – Sequence trace data provides a rough distribution of the relative frequency of each base at a given position. The medical implications of an acquired mutation rest not only on which mutations are present, but how frequent they are in the given population. In the more general population frequency determination problem, we are given an observed fraction of each base at each position and have to reconstruct the frequency of each mutation in the population. Formally, our problem is:

Input: A wildtype sequence S, a set V of allowable variations on S, and an experimental profile, consisting of the observed relative frequency for each $s \in \Sigma$ at each of n positions.

Output: The population frequency of each variation $v \in V$ that best matches the observed input profile.

1.2 Organization

Our paper is organized as follows. In Section 2, we consider the computational complexity of several variants of the sequence deconvolution problem, establishing natural classes of mutations for which reconstruction is NP-complete. Section 3 presents the practical algorithms which we developed for use in our implementation. This implementation was used to generate simulation results, described in Section 4. Analysis and experiments for the special case of reconstructing SNPs appears in Section 5, followed by future work in Section 6.

2. MUTATION DECONVOLUTION: SPECIAL CASES

Here we treat the mutation deconvolution problem from the strict standpoint of algorithmic theory. We demonstrate the boundary between classes of mutations are algorithmically easy to reconstruct and more general classes that are hard. Further, we categorize certain classes of mutations which mask each other in trace data, thus being inherently unreconstructable.

We consider the mutation deconvolution problem for increasingly general sets of allowable variations of substitutions, deletions, and insertions.

2.1 Substitution Sets

The simplest class of substitutions just change single bases. There are only $(\alpha - 1)n$ single-base variation sequences. If only single character substitutions are allowed, there is only one possible way to create each additional profile character, and hence the algorithmic reconstruction problem is trivial.

A more interesting and general class of mutations allows larger substrings. Here, each element of the set of allowable variations substitute one substring of length k in S with a different k-substring. We assume all $n\alpha^{2k}$ possible k-length substitutions exist in V.

THEOREM 1. *Mutation deconvolution for k-length substitutions can be solved in $O(\alpha n)$ time.*

PROOF. This can be solved optimally using a left to right greedy algorithm. Start with the leftmost uncovered profile character, and select the string which covers them and as large a set of other uncovered profile characters as possible. Since all k-string replacements are available, no decision precludes any other covering option. □

The incomplete substitution set problem is a restricted variant of the complete substitution set problem. Here, substitutions of k-substrings are made using only possible mutations from the set $\bar{M}$.

THEOREM 2. *The incomplete substitution set problem is NP-complete and is hard to approximate within a* log *factor.*

PROOF. A solution to the problem can clearly be verified in polynomial time. To show hardness, we reduce the set cover problem to the incomplete substitution sets problem; this reduction maintains hardness of approximation.

The set-cover problem is defined as follows. Given an integer set $N = \{1, \ldots, n\}$ and a set M of m subsets of N, find the smallest subset of M such that $\bigcup_{p \in M} p = N$. Given a set-cover instance (N, M), we introduce a character-sequence representation for each of the elements of M. For all integers i from 1 to n, if $i \in p$ append '*' to s, otherwise append '-' to s. Thus, for $n = 5$ the subset $p = \{1, 3\}$ is represented as '*-*--'.

The wildtype S consists of n consecutive '-'s; $a_i =$ '-' for all basepair positions of S. The profile consists of the wildtype plus a '*' in each column. $\bar{M}$ consists of the character-sequence representations of the subsets in M. A mutation $\hat{m}_k$ in $\hat{M}$ is constructed using a substitution of the entire wildtype S with an n-substring in $\bar{M}$.

There is a clear one-to-one correspondence between the set cover instance and the encoding. Each mutation is created using a substitution of the wildtype with a substring in $\bar{M}$, and each substring in $\bar{M}$ represents a subset in M. The reduction is correct and maintains hardness of approximation. □

We note that the greedy heuristic for set cover gives an $O(\lg |V|)$ factor approximation for this and indeed all variations of mutation deconvolution. The greedy heuristic identifies all elements of V consistent with the profile. While there are uncovered characters in the profile select the survivor which covers the most remaining profile characters.

2.2 Deletion Sets

In single character deletion, each possible mutation differs from wildtype S in the deletion of one character position. Thus there are n possible variation sequences.

We claim that any realizable profile can be covered using S and at most one other sequence. Observe that any two variants are identical to the left of the first deletion and to the right of the second deletion. Also observe that S is identical to both sequences to the left of their deletion. A generalization of this argument yields:

LEMMA 1. *Let V' be a minimum size solution for the mutation deconvolution problem. Then V' does not contain two deletion mutations of the same length k for any $1 \leq k \leq n$.*

Thus to cover the rightmost uncovered character of the profile, we can select the variant which has the leftmost deletion which is consistent with the rest of the profile. Note that if this does not completely cover the profile, then the profile cannot be covered, as all other consistent profiles share a prefix with S. Hence:

THEOREM 3. *The mutation deconvolution problem for k-length deletions can be solved in $O(\alpha n)$ time.*

In the *single range deletion* problem, each variation differs from S in the deletion of one contiguous subsequence, and all possible deletions are included. Thus there are $\binom{n}{2}$ possible variation sequences in V.

THEOREM 4. *The single range deletion problem is NP-complete for $|\Sigma| = m + 2$.*

PROOF. The problem is clearly in NP. Given a set-cover instance (N, M) we construct an instance of the single-run-deletion problem which uses a character-alphabet Σ of size $m + 2$; we later show how to encode this using only a three letter alphabet that corresponds to the nucleotide set {A,C,T}.

We begin by introducing a character sequence representation for the elements of M. Given a subset $p \in M$, represent p with a character sequence of length $n + 1$. Start with an empty character-sequence s, and construct from left to right. For all integers i from 1 to n, if $i \in p$ append '$*$' to s, otherwise append '$-$' to s. Conclude the construction by appending '#' to s. Thus, for $n = 3$ the subset $p = \{1, 3\}$ is represented as "$*{-}*\#$". We call '$-$' a place holder, '$*$' a marker and '#' a terminator.

Details of the construction appear in the full paper, but an example is given in Figure 2.

This construction reduces the input of a set-cover problem instance (N, M) to the input of a single-run-deletion problem instance with character alphabet $m + 2$. Any solution to the single-run-deletion problem instance includes $m - 1$ mutations that cover the elements of the alternating sets at positions greater than $n+1$. All other mutations in the solution must be used to cover the elements of the alternating sets a_1 to a_n. These mutations can be distinguished by examining the terminating positions of the deletions used to create them. Mutations created using a deletion that terminates left of position $2m(n + 1)$ are discarded. Each of the remaining mutations corresponds to choosing a single subset in the set-cover problem. A part of the chosen subset's representation must follow immediately after the deletion terminating position, and no part of any other subset's representation may appear in a position $x < n$. □

In the full paper we show how to encode each character of Σ as a unique $(m + 3)$-long sequence of nucleotides from {A,C,T}.

THEOREM 5. *The single-run deletion problem is NP-complete even for alphabets of size 3.*

2.3 Insertion Sets

The case of insertion mutations is similar to that of deletion mutations, since the leftmost insertion masks the shift of all insertions further right. Thus the single base insertion problem can be solved by including the insertion mutation defining the leftmost alternative character to the wildtype. Each remaining uncovered alternative character must be covered by a distinct insertion of the prescribed character. This yields a minimum covering if the union is consistent with the profile; otherwise no such covering exists. An extension of this algorithm can be used to find the minimal covering of any realizable profile with insertions of exactly length k in polynomial time.

THEOREM 6. *The mutation deconvolution problem for k-length insertions can be solved in $O(nk)$ time.*

In the *single range insertion of maximum size k* problem, each variation differs from S in the insertion of a single contiguous subsequence whose length is less than or equal to k. Thus there are $O(n\alpha^k)$ possible variations in V.

THEOREM 7. *The single range insertion of maximum size k problem is NP-complete for $|\Sigma| = 4$.*

PROOF. The problem is clearly in NP. Given a set cover instance (N, M) we construct an instance of the single range insertion of maximum size k problem which uses a character-alphabet of size 4.

Introduce a character sequence representation for the elements of M using the alphabet $\{*, -, \#, 1\}$. For each $p \in M$ we construct a subsequence s_p that consists of a series of '*' and '-' characters. If $i \in p$ then the ith character of s_p is a '*', otherwise, it is a '-'. A place holder subsequence s_- consists of n '-' characters.

The wildtype S consists of m copies of the following string: $\#s_1\#s_2\#\ldots\#s_m\#s_-$

To create the rest of the profile, a $(m + 1)$st repetition of the subset/place holder subsequences is added starting at position $(m + 1)(n + 1)$ (just off the end of the

```
    N={1,2,3,4}      M={{1,2},{2,3},{3,4}}

S:  ----#----#----#----#-**-2--**3**--#-**-#--**#
    **** -**-2--**3**--#-**-#--**#
         --**3**--#-**-#--**#
```

Figure 2: An example for the construction of a character-alphabet single-run-deletion problem instance from a set-cover problem-instance (N, M). A set of alternating characters a_i is composed of the characters below the ith position of S, given that alternating character sets are not multi-sets and may not include $s(i)$.

```
        N = {1, 2, 3, 4}      M = {{1, 2}, {2, 3}, {3, 4}}

S:  #**--#-**-#--**#----#**--#-**-#--**#----#**--#-**-#--**#----
                                                                #**--#-**-#--**#----
    1--**1*--*1**--1****1--**1*--*1**--1****1--**1*--*1**--1****1--**1*--*1**--1****
                    1111                1111                1111                1111
```

Figure 3: An example for the construction of a character-alphabet single-run-insertion problem from a set-cover problem-instance (N, M). See text for details.

sequence). Then, at each position, for every '*' in the profile, a '-' is added to the profile (and vice versa), and for every '#' in the profile, a '1' is added to the profile (and vice versa). Finally, n '1's are added to the profile beneath each copy of s_{-}. Figure 3 shows this construction for a sample set cover problem.

Note that all legal shifts must be of $c(m+1)$ characters, and all such shifts are allowed (for $c \leq n+1$). All '1's in the profile will have to be covered by the inserted subsequence instead of by the shifted wildtype since there are no '1's in the wildtype.

Since the profile is $(n+1)(m+1)$ longer than the wildtype, that is the maximum length of any insertion, meaning that at least $m+1$ insertion mutations will be needed to cover the '1's in the profile, and none of them will be able to cover the '*'s beneath the s_{-}. By choosing these $m+1$ insertions appropriately, it is possible to cover the rest of the profile.

A solution to the set cover problem can be made into a solution to insertion problem. If the ith subset is in the solution to (N, M), then the insertion problem contains an insertion of length $(n+1)*(i-1)$ starting at location 1. This insertion shifts the relevant subsequence into the place holder subsequence.

Other ways to cover this profile are more expensive. Details appear in the full paper. For example, it is possible to cover one or more '*'s in the place holder subsequence by inserting them explicitly. However, since there are $m+1$ copies of the place holder subsequence, you need $m+1$ insertions to do this (in addition to the m+1 insertions to cover the '1's), which means that any such solution is not a minimal one. □

3. ALGORITHMS AND IMPLEMENTATION

In this section, we describe the algorithms we implemented for each of the three problems described in Section 1.1. Our experiments with this implementation are reported in Section 4.

3.1 Base Calling

The base calling problem on actual gels is complicated by a variety of technology-dependent issues. The contribution of this paper is in higher-order determination of sequences from mixed populations, so we assume that the sequencing machine returns an observed frequency of each base at each position.

Here we describe how we performed base-calling in our simulation. Given the observed frequency $F(i, j)$ for base i at a given position j, we classify it based on thresholds t_{lo} and t_{hi}:

$$C(i,j) = \begin{cases} \text{Present} & F(i,j) > t_{hi} \\ \text{NoCall} & t_{lo} \leq F(i,j) \leq t_{hi} \\ \text{Absent} & F(i,j) < t_{lo} \end{cases}$$

Each element v of the set of candidate mutations V can be easily tested to see whether it is consistent with the set of called bases. A mutation v is considered consistent if for all i, $C(i, v(i)) \neq$ Absent. In other words, a mutation is consistent if each of its basepairs correspond to a Present or NoCall entry.

3.2 Mutation Deconvolution

We solve the mutation deconvolution problem as a variant of set cover problem consists of finding a minimal set of consistant mutations such that for every $C(i, j)$='Present', there is some $v \in V'$ such that $v(i) = j$.

Instead of using heuristics to approximately solve set cover, we explored the power of exhaustive search, specifically a DFS A* search. As each 'Present' value in the called-base array is accounted for, it is marked covered. A 'score' is defined that evaluates a given mutation and returns the number of 'Present' values that it will convert to covered if it is added to the solution set.

Mutations are added to the solution in decreasing order of the number of remaining 'Present' bases they will cover. An aggressive pruning scheme is used for early termination: if the best solution so far covers p mutations and we have already added q mutations to our solution, then score(v) must be at least $score(C)/(p-q)$ to be considered at this level. Below this threshold, the search tree can be pruned.

3.3 Population Frequency Determination

The third stage of the algorithm takes the solution of the set cover problem and the observed frequency matrix F. From this data it creates a system of up to $4|s|$ linear equations (where $|s|$ is the length of the wildtype). The linear equation for basepair i, value j, is

$$\sum_{k=1}^{|M|} w_k c(i,j,k) = F(i,j) \qquad (1)$$

where

$$c(i,j,k) = \begin{cases} 1 \text{ if } m_k(i) = j \\ 0 \text{ otherwise} \end{cases} \qquad (2)$$

Many of these equations will be degenerate, i.e. $c(i,j,k) = 0$ for all mutations. Specifically, if $C(i,j) =$'Absent', the equation is guaranteed to be degenerate, and if $C(i,j) =$ 'NoCall', the equation may or may not be degenerate. All degenerate equations are discarded from the solution.

The resulting system of equations will be an overconstrained system of linear equations. This system can be written as $A\vec{w} = \vec{y}$ and can be solved by the left-multiplying the pseudo-inverse $A^* = A^t(AA^t)^{-1}$ on both sides of the equation. The resulting weight vector w contains the weights that minimize the squared error.

One quirk of the least squares solution is that it does not enforce the constraint that all weights must be non-negative. To get a least-squares solution while enforcing the non-negativity constraint, we may reformulate our linear system as a quadratic program with less than $4|s|$ constraints and as many dummy variables. If we're interested in a least-displacement solution (i.e. minimize $|error|$ instead of $error^2$), we can make it into a linear program with less than $4|s|$ constraints and twice as many dummy variables as constraints. Both of these alternatives are more costly to solve.

4. EXPERIMENTAL RESULTS

We performed a series of computational experiments to evaluate how accurately multiple mutations can identified in a mixed population subject to experimental error, and to measure the performance of our algorithms. Below, we discuss the sources of our test data and our results.

4.1 Test Data

To accurately simulate a real-world diagnostic application, the library of mutations we used in our study were derived from a large database of p53 mutations known to cause cancers in humans. We used test data from the World Health Organinzation's International Agency for Research on Cancer, Lyon, France, specifically version R5 (June 2001) of [13], which appears to be the largest p53 database available.

We limited our experiments to exon 4 mutations, since this is the largest exon with any significant number of mutations and in principle the most challenging computationally. There are 167 distinct substitutions, 22 distinct insertions, and 76 distinct deletion mutations to exon 4 in this database. The database entries for the insert mutations contained only information about where it occurred and how long the insertion sequence was, but not the inserted sequence itself. So each insertion mutation in the database was modified by inserting a random sequence of the correct length at the correct location. (i.e. each insert mutation had a sequence chosen at random, but those sequences were fixed and known before the problem was started). The vast majority of insert/delete mutations are short (length ≤ 5), although the longest reported deletion has length 278.

At each noise level we considered, 50 random problems were created. Six mutations were chosen at random for each problem. For runs requiring $k < 6$ mutations, the first k mutations were selected. For a noise level n and a m mutation problem, the weight was chosen to be between $n/2$ and $0.6/m$ The base-frequency matrix F is then generated by tabulating which sequences contribute to which basepairs and adding uniform noise centered around the expected frequency.

4.2 Results

We ran extensive simulations on reconstructing from 1 to 6 mutations, subject to noise levels from 1% to 30% relative error. Each time point is represented by 50 solved problems.

Our system seeks to explain the observed data in terms of the fewest mutations possible. There may be multiple solutions of optimal cardinality, complicating the interpretation of the quality of our results. We summarize our results in four graphs:

- *Correct Mutations Identified* – Figure 4(a) reports the likelihood that a mutation in the correct answer was found by at least one of the 'optimal' solutions found by the program. These results demonstrate that we can identify at least 90% of present mutations even subject to a data error rate of 5% with up to 6 mutations.

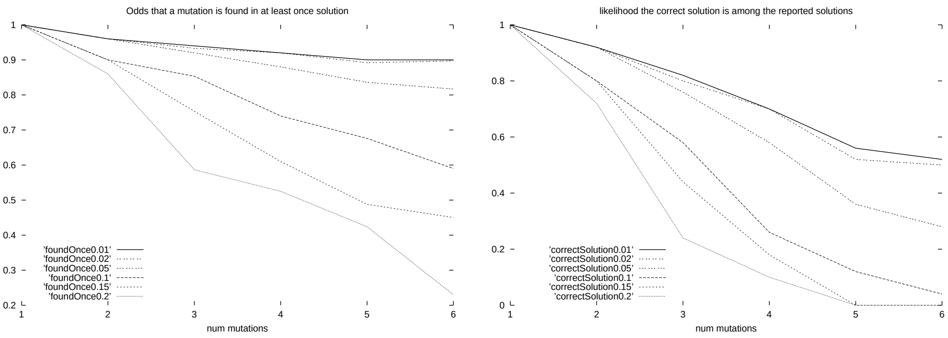

Figure 4: The likelihood that (a) a correct mutation and (b) all correct mutations are found by one of the reported solutions. The noise introduced into the experiments varies between 1% and 20%.

- *Correct Mutation Set Identified* – A stronger condition requires that we identify the complete set of mutations present. Figure 4(b) returns the likelihood that an optimal solution found by the program is identical to the correct answer. Our accuracy degrades with observed error and the number of mutations, but we can identify the full solution of up to 4 mutations roughly 60% of the time given an observed error of up to 5%.

- *Correlation with Observed Frequency* – Figure 5(a) measures the accuracy with which we reconstruct the weight of the mutations and wildtype. The weights of the correct answer and optimal solutions were converted to vectors and the correlation of these vectors computed as the cosine of the angle between them. When the system reported multiple optimal solutions, each was weighted equally.

  We demonstrate a high correlation for up to 4 mutations even subject to an observed error of up to 30%. For error rates of 5% we can accurately reconstruct the distribution even with 6 mutations.

- *Correlation with Observed Frequency When Mutations are Known* – Figure 5(b) measures the accuracy with which we reconstruct the weight of the mutations and wildtype if the set-cover problem returns the correct mutations. Correlations were computed as in the previous section. The correlation for all cases is very high, and when the noise is less than 0.1, the correlation is within 0.05% of the correct solution.

5. SNP GENERATION AND ANALYSIS

A single-nucleotide polymorphism (SNP) is a mutation that differs from a wildtype by a single substitution in one base position. An important current problem in genomics is cataloging all SNPs in a given population. As discussed in Section 2.1, it is easy to detect SNPs when sequencing samples sequentially. Here, we show how to employ a pooling strategy to detect SNPs in multiple individuals through a single sequencing run. This results in a significant increase in sequencing throughput when compared with sequencing each individual seperately.

In our pooling strategy, we combine equal amounts of DNA from each of the m distinct invividuals, and amplify the region of interest using PCR. We then sequence the resulting mixture on a frequency-sensitive machine. Our analysis question is determining the conditions under which the peak resulting from an SNP is distinguishable from the background noise of the sensor.

5.1 Analysis

If we assume that at most one SNP occurs per base location, we can look at the odds that such a mutation will be detected. We assume that the noise of our detector is uniform noise $U(0, \epsilon)$ and the number m of individuals in the mix is known.

A SNP can be detected in two different ways: by a larger than normal measurement off the wildtype (v), or by a smaller than normal measurement on the wildtype (w). If $v > \epsilon$, then the only way that such a reading could occur is if there is a SNP. If we assume the total weight of the mixture is normalized to 1.0, a SNP in one individual will have a measurement of $1/n$ before the sensor error is added, and weight $1/n \leq v \leq 1/n + \epsilon$ afterwards. If $1/n > \epsilon$, the measurements from SNPs will be measureably different than those resulting from noise, and all SNPs will be detected correctly.

In the case that $1/n < \epsilon$, there is a chance that a SNP will have a measurement that is less than ϵ, just as there is a chance that the background noise will exceed $1/n$. The odds of such an occurrence are

$$P(v < \epsilon) = \frac{\epsilon - 1/n}{\epsilon} = 1 - \frac{1}{\epsilon n} \qquad (3)$$

However, even if $v < \epsilon$, it still may be possible to detect the presence of a mutation because of the lower w value. Specifically, $w = 1.0 - 1/n + U(0, \epsilon)$, so

$$P(w > 1) = \frac{\epsilon - 1/n}{\epsilon} = 1 - \frac{1}{\epsilon n} \qquad (4)$$

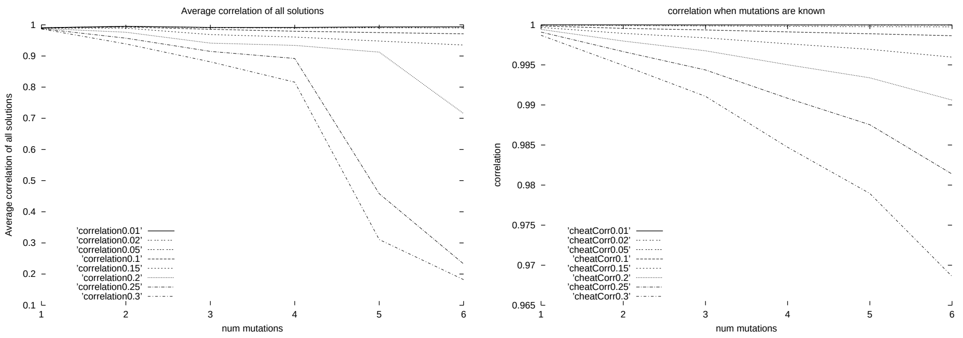

Figure 5: Correlation of reconstructed and original population frequencies as mutation count and error rate, using (a) experimentally reconstructed mutations, and (b) the actual mutation set – note vertical axis scale. The introduced noise varies between 1% and 30%

This means that the odds of missing a SNP completely is

$$P(v < \epsilon) \cdot P(w > 1) = (1 - \frac{1}{\epsilon n})^2 = 1 - \frac{2}{\epsilon n} - \frac{1}{\epsilon^2 n^2} \quad (5)$$

However, w being too small isn't enough information to identify which of the three off-wildtype SNPs has occurred.

The process for detecting multiple SNPs in a single basepair is basically the same as detecting a single SNP. An elevated v value means a SNP is present, as does a depressed w value. However depressed w will be less likely to disambiguate the results. As shown in Figure 6, a system with two or more SNPs is more likely to be ambiguous because the noise is more likely to interfere than with a single SNP.

5.2 Simulation Results

We performed a set of experiments to confirm the mathematical analysis, as well try to understand how the system behaves in unusual cases, such as multiple strains of data having the same mutation.

For the simulations below the all candidates considered were single-base mutations. A mixture of m different strains was created where strains were likely to be mutated with both probability 0.01 and 0.001. Noise drawn from $U(0, \epsilon)$ was added to the measurements to simulate sensor error. The measurement was then checked to see if it was consistent with more than one set of mutations. In that case, the measurement was declared 'ambiguous'. For example, an ambiguous measurement might be one where the original mixture contained 1 a-c mutation and 2 a-g mutations, but the measurement was also consistent with 1 a-c and 3 a-g mutations.

Figure 7 shows that even in the case of large amounts of sensor error, it is possible to reliably simultaneously detect SNP variations in multiple sequences. When the sensor noise is reduced, it may be possible to test more than 100 sequences while detecting all SNP mutations.

6. FUTURE WORK

The most obvious future work is measuring how well the set-cover and SNP algorithms work on real data instead of synthetic data. Probably the most difficult part of this will be working with a more realistic noise model. Uniform noise models are good for analysis, but are probably not representative of the noise in real systems. While we expect our techniques to work for other noise models, they have yet to be tested.

One area that is gaining importance is the detection of haplotypes instead of SNPs. Haplotypes are a combination of alleles of closely linked loci that are found in a single chromosome, tend to be inherited together, and in some cases can be used to determine genetic traits. Current research suggests that haplotypes may be more important than SNPs in determining genetic predisposition, and suggest creating a map of all human haplotypes [3].

The techniques in section 5 do not maintain any correlation information between SNPs, so it is not possible to directly obtain haplotypes in the same way. Also, all the results are based on single runs of a sequencer. Since the loci for haplotypes is typically larger than the run length of a sequencer (10k+ basepairs vs. 1k basepairs), it is often not possible to get information about a haplotype from a single run. To deal with this, the haplotyping community has developed many techniques to detect haplotypes without having to do multiple-run sequencing [12, 7, 15]. It is not clear whether section 5 could be used in conjuction with these techniques.

However, a modification of our method can be used to identify multiple SNPs occurring in the same individual. By combining differing amounts of DNA from each individual, it may be possible to use the frequency output of the machine to correlate SNPs. However, since the noise of this output is high, it would mean that only smaller pools can be simultaneously sequenced. A second alternative would be to create multiple, redundant pools of DNA so that each piece of DNA is sequenced

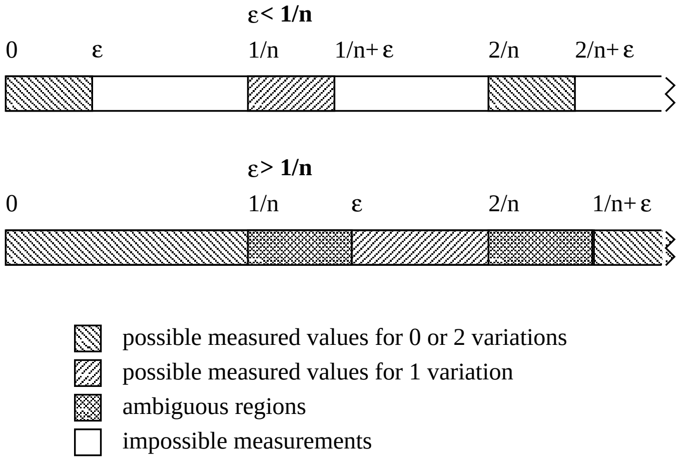
ε< 1/n
0
ε
1/n
1/n+ ε
2/n
2/n+ ε
ε> 1/n
0
1/n
ε
2/n
1/n+ ε
▧ possible measured values for 0 or 2 variations
▨ possible measured values for 1 variation
▦ ambiguous regions
□ impossible measurements

Figure 6: A graphical representation of the effect of noise on observed measurements in the model. While $\epsilon < 1/n$ there is no noise to obscure SNPs. Once $\epsilon > 1/n$, there are some measurements that are ambiguous, i.e. it's not possible to deconvolve them with certainty.

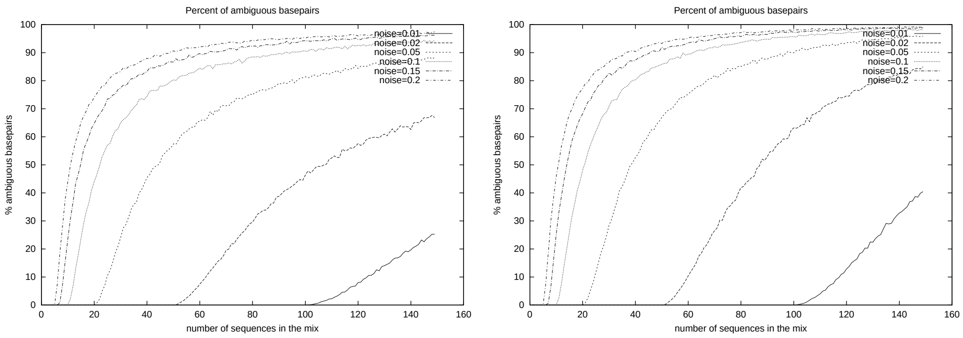

Figure 7: The likelihood that a mixture of m sequences under sensor error ϵ will result in a measurement that could be ambiguouosly interpreted, (a) for SNP frequencies of 1/1000 bases, and (b) for SNP frequencies of 1/100 bases.

more than once, and correlate the output between to determine likely haplotypes. However, this would also reduce the gains in throughput, and relies on the assumption that SNPs are very rare which is not always the case.

Acknowledgment

We thank Dr. Vera Gorfinkel of BioPhotonics Corp. for introducing us to this problem and useful discussions.

7. REFERENCES

[1] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachmann, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. In R. Shamir, S. Miyano, S. Istrail, P. Pevzner, and M. Waterman, editors, *Proceedings of the 4th Annual International Conference on Computational Molecular Biology (RECOMB-00)*, pages 54–64, N.Y., Apr. 8–11 2000. ACM Press.

[2] C. Beroud and T. Soussi. p53 gene mutation: software and database. *Nucleic Acids Research*, 26:200–204, 1998.

[3] M. Daly, J. Rioux, S. Schaffner, T. Hudson, and E. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29(2):229–232, October 2001.

[4] G. Denisov, D. Ho, M. Mettler, J. Candlin, and T. Hunkapiller. Tracetuner – next generation base calling. www.paracel.com, September 2000.

[5] J. DeRisi, V. Iyer, and P. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, Oct. 24, 1997.

[6] B. Ewing, L. Hillier, M. Wendl, and P. Green. Base-calling of automated sequencer traces using phred. i. accuracy assessment. *Genome Research*, 8:175–185, 1998.

[7] L. Excoffier and M. Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol. Biol. Evol.*, 12(5):921–927, 1995.

[8] S. Fodor, J. Read, M. Pirrung, L. Stryer, A. Lu, and D. Solas. Light-directed, spatially addressable parallel chemical synthesis. *Science*, 251:767–773, 1991.

[9] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Caasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[10] V. Gorfinkel and S. Luryi. Method and apparatus for identifying fluorophores. US Patent 5784157, July 21, 1998.

[11] I. S. M. W. Group. A map of human genome sequence variation containing 1.4 million snps. *Nature*, 409:928–933, 2001.

[12] M. E. Hawley and K. K. Kidd. Haplo: a program using the em algorithm to estimate the frequencies of multi-site haplotypes. *J. Hered.*, 86:409–411, 1995.

[13] T. Hernandez-Boussard, P. Rodriguez-Tome, R. Montesano, and P. Hainaut. Iarc p53 mutation database: a relational database to compile and analyze p53 mutations in human tumors and cell lines. *Hum. Mutat.*, 14(1):1–8, 1999.

[14] R. Tang, P. Wang, H. Wang, J. Wang, and L. Hsieh. Mutations of p53 gene in human colorectal cancer: Distinct frameshifts among populations. *Int. J. Cancer*, 91:863–868, 2001.

[15] S. A. Tishkoff, A. J. Pakstis, G. Ruano, and K. K. Kidd. The accuracy of statistical methods for estimation of haplotype frequencies: An example from the cd4 locus. *Am. J. Hum. Genet.*, 67(2):518–522, 2000.

[16] D. Walther, G. Bartha, and M. Morris. Basecalling with lifetrace. *Genome Research*, 11:875–888, 2001.