On Maximum Likelihood Reconstruction over Multiple Deletion Channels

Sundara Rajan Srinivasavaradhan, Michelle Du, Suhas Diggavi and Christina Fragouli*

***Abstract*—The problem of reconstructing a sequence when observed through multiple looks over deletion channels occurs in "de novo" DNA sequencing. The DNA could be sequenced multiple times, yielding several "looks" of it, but each time the sequencer could be noisy with (independent) deletion impairments. The main goal of this paper is to develop reconstruction algorithms for a sequence observed through the lens of a *fixed* number of deletion channels. We use the probabilistic model of the deletion channels to develop both symbol-wise and sequence maximum likelihood decoding criteria, and algorithms motivated by them. Numerical evaluations demonstrate improvement in terms of edit distance error, over earlier algorithms.**

***Index Terms*—deletion channels, sequence decoding, MAP symbol decoding, trace reconstruction**

I. Introduction

The problem of reconstructing (or "decoding") a sequence when observed through multiple looks over deletion channels occurs in "de novo" DNA sequencing, *e.g.,* see models and analysis for recent nanopore sequencers [1], [2], [3]. The DNA could be sequenced multiple times, yielding several "looks" of it, but each time the sequencer could have (independent) deletion impairments [1], [2]. The main goal of this paper is to develop reconstruction algorithms for a sequence observed through the lens of a *fixed* number of deletion channels.

A similar problem has received attention in CS theory and discrete mathematics, under the name "trace-reconstruction" [4], [5], [6], [7], [8]. The main question these papers address is what is the number of traces needed for *perfect* reconstruction of an input sequence. They show that the required number of traces (or looks) through independent deletion channels grows with the input length, either exponentially in the worst case or sub-polynomially in the average case. In contrast, we are interested in the case where we have a fixed number of deletion channels, motivated by a finite number of reads of a sequence. Moreover, we do not aim for perfect reconstruction, but a maximum likelihood decoder for the input sequence, motivated by a more traditional information-theoretic decoding criterion. [9] (also see references therein) also considered decoding over channels with synchronization errors, but as far as we know has not examined multiple parallel channels.

There has also been extensive work on sequence assembly, where multiple short reads are used to reconstruct an original sequence. This falls into two categories. One where many algorithms have been developed and implemented for assembly with noisy reads, but without theoretical performance guarantees for insertion/deletion errors in the reads ([10] and references therein). The other is information-theoretic work to examine the number of reads to assemble a sequence from noiseless reads [11]. In contrast, we examine the case of "de novo" sequencing with multiple reads of the *same* sequence through deletion channels, motivated by multiple reads of a nanopore sequencer.

In this paper[1], we formulate maximum likelihood decoding criteria for reconstruction through multiple deletion channels, which we believe has not been examined before. We develop algorithms for a *fixed* (and small) number of traces, which optimize symbol-wise reconstruction when the original sequence length is known as well as sequence reconstruction when the original length is unknown. Technically, we use a tool called the infiltration product, which we believe has not been used for the reconstruction problem over deletion channels before, and which enables development of an equivalent representation of the problem that may be of independent interest. For the sequence reconstruction problem, we show that for sufficiently small deletion probability, the shortest common supersequence decoder has significantly higher likelihood, making it a good candidate for decoding. First evaluation results indicate that in terms of edit distance error, our approach can outperform earlier algorithms such as the majority voting methods [5].

The paper is organized as follows. Section II presents the model and definitions; Section III proves an equivalence based on infiltration products; Section IV develops exact symbol-wise MAP/ML decoding criteria and algorithms for multiple traces, when the input length is known; Section V develops the sequence estimation criterion and algorithms; and Section VI presents evaluation results.

II. Notation and problem formulation

Consider a sequence X passed through t independent deletion channels as shown in Fig. 1. When the sequence is passed through a channel, each bit is independently deleted with probability δ. This is a simple model that captures the process of the same DNA sequenced t times through a nanopore sequencer, with each read modeled for simplicity as a deletion channel[2]. The goal is to estimate X from the *traces* $\{Y_i\}$s. It is clear that each Y_i is a subsequence of X.

We formulate the reconstruction problem as follows. When we exactly know the length n of the original se-

*The authors are with the Department of ECE, UCLA. E-mails: {sundar, michelleruodu,suhas.diggavi,christina.fragouli}@ucla.edu. S.R.S. acknowledges Guru-Krupa fellowship received during Jan-March 2018, and M. Du was supported through an undergraduate NSF REU during summer 2017. This work was supported, in part, by NSF grants 1705077 and 1514531.

[1]A longer version of this work is available at [12].

[2]As seen in [1],[2] there are more complicated effects of the nanopore reader not captured in this simple representation.

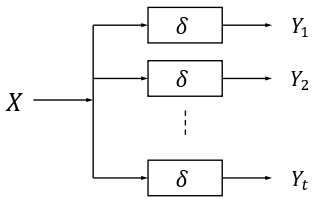

Fig. 1. The input sequence X is passed through t independent deletion channels. We aim to estimate X from the Y_is.

quence, we can compute the symbolwise maximum aposteriori estimate as $\hat{X}_{sm} = \hat{X}_1\hat{X}_2...\hat{X}_n$, as

$$\hat{X}_i = \arg\max_{a\in\mathcal{A}} \Pr\left(X_i = a|Y_1, ..., Y_t\right). \tag{1}$$

We calculate the maximum-likelihood sequence estimate as

$$\hat{X}_{ml} = \arg\max_{h} \Pr\left(X = h|Y_1, ..., Y_t\right). \tag{2}$$

which can be used even when the input length is unknown.

Notation: Calligraphic letters refer to sets and capitalized letters correspond to random variables. Let $\mathcal{A}$ be the set of all symbols. We will focus on the case where $\mathcal{A} = \{0, 1\}$, though our methods extend to larger sets. Define $\mathcal{A}^n$ to be the set of all n-length sequences, $\mathcal{A}^*$ to be the set of all finite length sequences with symbols in $\mathcal{A}$. For a sequence f, $|f|$ denotes the length of f. Given sequences f and g in $\mathcal{A}^*$, the number of subsequence patterns of f that are equal to g is called the *binomial coefficient* of g in f and is denoted by $\binom{f}{g}$. When the alphabet $\mathcal{A}$ is of cardinality 1, $\binom{f}{g} = \binom{|f|}{|g|}$, the classical binomial coefficient with their respective lengths as the parameters. This definition hence could be thought of as a generalization of the classical binomial coefficients. We will denote by e the sequence of length 0, and define $\binom{f}{e} \triangleq 1 \ \forall \ f \in \mathcal{A}^*$.

Edit distance: The edit distance $d_e(f, g)$ measures similarity between two sequences of possibly different lengths [13]. $d_e(f, g)$ is the minimum number of operations needed to transform f to g, where the permitted operations are insertion, deletion or substitution of a symbol. In this work, we quantify the performance of algorithms in Section VI using the edit distance metric.

Edit graph (defined in [14]): We now define an *edit graph* where given two sequences $f$and g, every path on the edit graph yields a supersequence h of f, g, where h is "covered" by f, g – each symbol of h comes from either f or g or both. For f and g in $\mathcal{A}^*$, we form a graph $\mathcal{G}(f, g)$ with $(|f| + 1) \times (|g| + 1)$ vertices each labelled with a distinct pair $(i, j), 0 \leq i \leq |f|, \ 0 \leq j \leq |g|$. A directed edge $(i_1, j_1) \rightarrow (i_2, j_2)$ exists iff at least one of the following holds: 1) $i_2 - i_1 = 1$ and $j_1 = j_2$, or 2) $j_2 - j_1 = 1$ and $i_1 = i_2$, or 3) $i_2 - i_1 = 1$, $j_2 - j_1 = 1$ and $f_{i_2} = g_{j_2}$, where f_i is the i^{th} symbol of the sequence f.

Let $p = ((i_1, j_1), (i_2, j_2), ..., (i_m, j_m))$ be a path in $\mathcal{G}(f, g)$. We define $w(p)$ to be the sequence corresponding to the path, i.e., $w(p) = x_1x_2...x_{m-1}$ where $x_k = f_{i_{k+1}}$ if $j_k = j_{k+1}, x_k = g_{j_{k+1}}$ if $i_k = i_{k+1}, x_k = f_{i_{k+1}}$ else. Intuitively, $w(p)$ is formed by

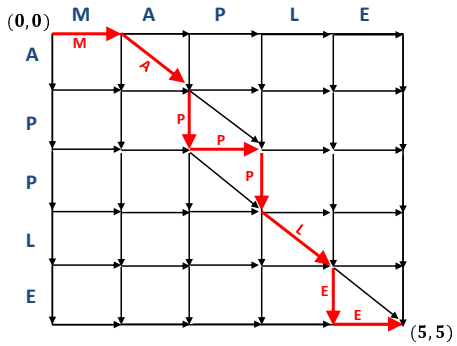

Fig. 2. Edit graph $\mathcal{G}$('APPLE','MAPLE'). An easy way to think about this is to write down f vertically with each symbol aligned to a vertical set of edges and g horizontally likewise. A diagonal edge in a square exists if the corresponding f and g symbols are equal. The thick red arrows form a path and the symbols next to it form the corresponding sequence.

appending symbols in the following way: append the corresponding f symbol for a vertical edge, g symbol for horizontal edge, and f or g symbol for diagonal edge (see example Fig. 2).

Infiltration product: Using the edit graph we can construct the set of possible supersequences $\mathcal{S}(f, g)$ of f, g which are covered by it. Clearly multiple paths could yield the same supersequence and we can count the number of distinct ways $N(h; f, g)$ one can construct the same supersequence h from f, g. We can informally define the *infiltration product* $f \uparrow g$ of f and g, as a polynomial with monomials the supersequences h in $\mathcal{S}(f, g)$ and coefficients $\langle f \uparrow g, h\rangle$ equal to $N(h; f, g)$. For example, let $\mathcal{A} = \{a, b\}$, then

$$ab \uparrow ab = ab + 2aab + 2abb + 4aabb + 2abab$$
$$ab \uparrow ba = aba + bab + abab + 2abba + 2baab + baba.$$

The definition of infiltration extends to two series via distributivity (precisely defined later), and to multiple sequences $f \uparrow g \uparrow h$ in the same way: $\langle f \uparrow g \uparrow h, w\rangle$ is the number of distinct ways of constructing w as a supersequence of f, g, h so that the construction covers w.

We now give a more formal definition of the infiltration product (see [15] for the equivalence of the two definitions and a more rigorous treatment). Denote $\mathbb{Z}$ to be the ring of all integers. A *series* with coefficients in $\mathbb{Z}$ and variables in $\mathcal{A}^*$ is a mapping of $\mathcal{A}^*$ into $\mathbb{Z}$. The set of these series is denoted by $\mathbb{Z}\langle\langle\mathcal{A}\rangle\rangle$. For a series $\sigma \in \mathbb{Z}\langle\langle\mathcal{A}\rangle\rangle$ and a sequence $w \in \mathcal{A}^*$, the value of σ on w is denoted by $\langle\sigma, w\rangle$, we refer to it as the coefficient of w in σ, and it is an element of $\mathbb{Z}$. Clearly for $f, g \in \mathcal{A}^*, \langle f, g\rangle = \mathbb{1}_{f=g}$. The following operations of sum and product(or concatenation) of two series $\sigma, \tau \in \mathbb{Z}\langle\langle\mathcal{A}\rangle\rangle$, turn $\mathbb{Z}\langle\langle\mathcal{A}\rangle\rangle$ into a ring [15]:

$$\langle\sigma + \tau, w\rangle = \langle\sigma, w\rangle + \langle\tau, w\rangle, \tag{3}$$
$$\langle\sigma\tau, w\rangle = \sum_{\substack{w=uv \\ u,v\in\mathcal{A}^*}} \langle\sigma, u\rangle\langle\tau, v\rangle \text{ for any } w \in \mathcal{A}^*. \tag{4}$$

A series is called a *polynomial* if all but finite number of its coefficients are zero. We now define a series called the

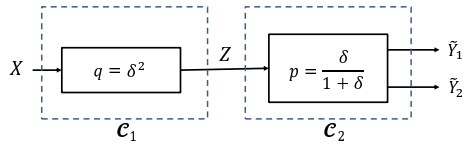

Fig. 3. The cascade of a single deletion channel with parameter $q = \delta^2$ with the remnant channel with parameter $p = \frac{1}{1+\delta}$.

infiltration product [15] as follows:

$$\forall f \in \mathcal{A}^*, \quad f \uparrow e = e \uparrow f = f. \tag{5}$$

$$\forall f, g \in \mathcal{A}^*, \quad \forall a, b \in \mathcal{A},$$

$$fa \uparrow gb = (f \uparrow gb)a + (fa \uparrow g)b + \mathbb{1}_{a=b}(f \uparrow g)a. \tag{6}$$

$$\forall \sigma, \tau \in \mathbb{Z}\langle\langle \mathcal{A} \rangle\rangle, \quad \sigma \uparrow \tau = \sum_{f,g \in \mathcal{A}^*} \langle \sigma, f \rangle \langle \tau, g \rangle (f \uparrow g). \tag{7}$$

The infiltration operation is commutative and associative, and infiltration of two sequences $f \uparrow g$ is a polynomial with variables of length (or *degree*) at most $|f| + |g|$ [15].

III. Equivalent Formulation

We establish an equivalence of the $t = 2$ deletion channels with another channel model that exploits the infiltration product. To do so, we first state a relation between the binomial coefficients and the infiltration product, which forms the backbone for many of our results.

Proposition 1. *For $h, f_1, f_2, ..., f_m \in \mathcal{A}^*$,*

$$\binom{h}{f_1}\binom{h}{f_2}...\binom{h}{f_m} = \sum_{w \in \mathcal{A}^*} \langle f_1 \uparrow f_2 \uparrow ... \uparrow f_m, w \rangle \binom{h}{w}.$$

The full proof of the proposition can be found in [12], we give an intuition here. We use induction to prove it. The main idea is as follows: consider a particular occurrence of $f_1, ..., f_m$ as subsequences of h. For this instance, the symbols in h can be divided into two sets: symbols which are "covered" by $f_1, ..., f_m$ and symbols which are not. w refers to the subsequence of h covered by $f_1, ..., f_m$, and of course, there are multiple coverings, as given by the infiltration product. Summing over all w's gives us the expression in the left hand side.

Consider now the channel in Fig. 3, where the input is first passed through a single deletion channel $\mathcal{C}_1$ and then through the remnant channel $\mathcal{C}_2$. Each bit passing through $\mathcal{C}_2$ is deleted in $\tilde{Y}_1$ but appears in $\tilde{Y}_2$ with a probability p and vice-versa with the same probability, and it appears in both with a probability $1 - 2p$. Equivalently, for a sequence $z \in \mathcal{A}^*$ and outputs $f, g \in \mathcal{A}^*$:

$$\begin{aligned}\Pr(\tilde{Y}_1 = f, \tilde{Y}_2 = g | Z = z) \\ = \langle f \uparrow g, z \rangle p^{2|z|-|f|-|g|}(1-2p)^{|f|+|g|-|z|}.\end{aligned} \tag{8}$$

The following lemma shows that in terms of the input-output distribution such a cascade is equivalent to the $t = 2$ independent deletion channel.

Lemma 1. *The $t = 2$ independent deletion channel and the cascade in Fig. 3 produce the same* $\Pr(Y_1, Y_2|X)$.

Proof. We need to prove that

$$\Pr(\tilde{Y}_1 = f, \tilde{Y}_2 = g | X = x) = \Pr(Y_1 = f, Y_2 = g | X = x)$$

when $q = \delta^2$ and $p = \frac{\delta}{1+\delta}$ or equivalently,

$$\begin{aligned}\Pr(\tilde{Y}_1 = f, \tilde{Y}_2 = g | X = x) \\ = \binom{x}{f}\binom{x}{g}\delta^{2|x|-|f|-|g|}(1-\delta)^{|f|+|g|},\end{aligned} \tag{9}$$

since it is easy to see that the RHS is equal to $\Pr(Y_1 = f, Y_2 = g|X = x)$. Now,

$$\begin{aligned}&\Pr(\tilde{Y}_1 = f, \tilde{Y}_2 = g | X = x) \\ &= \sum_{z \in \mathcal{A}^*} \Big[\Pr(Z = z | X = x) \times \\ &\qquad \Pr(\tilde{Y}_1 = f, \tilde{Y}_2 = g | Z = z, X = x) \Big].\end{aligned} \tag{10}$$

Expanding the probability terms and using the fact that $X - Z - \tilde{Y}_1\tilde{Y}_2$ forms a Markov chain the previous expression is

$$\begin{aligned}&\sum_{z \in \mathcal{A}^*} \left[\binom{x}{z} q^{|x|-|z|}(1-q)^{|z|} \right] \\ &\qquad \left[\langle f \uparrow g, z \rangle p^{2|z|-|f|-|g|}(1-2p)^{|f|+|g|-|z|} \right].\end{aligned}$$

When $q = \delta^2$ and $p = \frac{\delta}{1+\delta}$, this is equal to

$$\begin{aligned}&\sum_{z \in \mathcal{A}^*} \binom{x}{z} \langle f \uparrow g, z \rangle \delta^{2|x|-|f|-|g|}(1-\delta)^{|f|+|g|} \\ &\qquad \overset{(a)}{=} \binom{x}{f}\binom{x}{g}\delta^{2|x|-|f|-|g|}(1-\delta)^{|f|+|g|}\end{aligned} \tag{11}$$

where (a) follows from Proposition 1 for $m = 2$. □

For $t = 2$ the maximum likelihood estimate is

$$\begin{aligned}\hat{X}_{ml} &= \arg\max_x \Pr(Y_1 Y_2 | X = x) \\ &= \arg\max_x \binom{x}{Y_1}\binom{x}{Y_2}\delta^{2|x|},\end{aligned}$$

which can be expressed using Proposition 1 as,

$$\hat{X}_{ml} = \arg\max_x \sum_{w \in \mathcal{A}^*} \langle Y_1 \uparrow Y_2, w \rangle \binom{x}{w} \delta^{2|x|}. \tag{12}$$

We can also write the ML decoding criterion for the input of the remnant channel (Fig. 3), given same outputs as,

$$\hat{Z}_{ml,rem} \equiv \arg\max_z \quad \langle Y_1 \uparrow Y_2, z \rangle \left(\frac{\delta^2}{1-\delta^2} \right)^{|z|}. \tag{13}$$

The objective function here does not sum over all coefficients of $Y_1 \uparrow Y_2$ unlike for the independent deletion channels, and hence constructing the infiltration $Y_1 \uparrow Y_2$ gives a table look-up way to find $\hat{Z}_{ml,rem}$. Thus the infiltration proves its use in estimating the intermediate sequence Z, which will be fairly close to the actual input X for small δ. For very small δ, it would also be fair to expect that $\langle Y_1 \uparrow Y_2, z \rangle \left(\frac{\delta^2}{1-\delta^2} \right)^{|z|}$ is maximum when $|z|$ is of the lowest possible value; this motivates the heuristic of decoding via a shortest common supersequence of the traces that we term Alg. 4 in Section V.

IV. Symbolwise MAP Estimate

We examine symbolwise maximum-aposteriori (MAP) estimation when the length n of the sequence is known, and every sequence in $\mathcal{A}^n$ is equally likely. We first consider the case of a single deletion channel and derive an algorithm (Alg. 1) that is polynomial in n; the extension to $t > 1$ channels (Alg. 2) may not be polynomial time, and we thus propose Alg. 3 motivated by a criterion for approximate MAP estimation. We present the methods for $t = 2$, and $\mathcal{A} = \{0, 1\}$, but they extend for general $\mathcal{A}$ and t.

• $\mathbf{t = 1}$: The posterior symbol-wise probability is,

$$\Pr(X_i = a|Y) = \sum_{x \in \mathcal{A}^n | x_i = a} \frac{\Pr(x)\Pr(Y|X = x)}{\Pr(Y)}$$
$$= \frac{\delta^{n-|Y|}(1-\delta)^{|Y|}}{2^n \Pr(Y)} \sum_{x \in \mathcal{A}^n | x_i = a} \binom{x}{Y}$$

We compute $\sum_{x|x_i=a} \binom{x}{Y}$ using the following Lemma.

Lemma 2.

$$\sum_{f \in \mathcal{A}^n | f_i = a} \binom{f}{g} =$$
$$2^{n-|g|} \left(\frac{1}{2} \binom{n-1}{|g|} + \sum_{j|g_j=a} \binom{i-1}{j-1} \binom{n-i}{|g|-j} \right),$$

where index j is implicitly constrained to $[|g| + i - n : i]$.

Proof. The high level idea is to think of the LHS as the total number of ways to first place g in $|g|$ out of n spaces, such that the i^{th} space is always a, and filling out the remaining spaces. Let $\mathbf{s} = (s_1, ..., s_m), s_i \in [1 : n]$ *and* $s_i < s_j, \forall i < j$ denote the indices (in increasing order) corresponding to a particular m-length subsequence of an n-length sequence, and let $\mathcal{S}_m$ denote the set of all possible $\mathbf{s}$ of length m. Clearly, $|\mathcal{S}_m| = \binom{n}{m}$. Also let $f_{\mathbf{s}} \equiv f_{s_1} f_{s_2} ... f_{s_m}$. Now,

$$\sum_{f \in \mathcal{A}^n | f_i = a} \binom{f}{g} = \sum_{f|f_i=a} \sum_{\mathbf{s} \in \mathcal{S}_m} \mathbb{1}\{f_{\mathbf{s}} = g\}$$
$$= \sum_{\mathbf{s} \in \mathcal{S}_m} \sum_{f|f_i=a} \mathbb{1}\{f_{\mathbf{s}} = g\}$$
$$= \sum_{\mathbf{s}|i \notin \mathbf{s}} \sum_{f|f_i=a} \mathbb{1}\{f_{\mathbf{s}} = g\} + \sum_{\mathbf{s}|i \in \mathbf{s}} \sum_{f|f_i=a} \mathbb{1}\{f_{\mathbf{s}} = g\}.$$

To evaluate the first term, consider Fig. 4(a). For a fixed

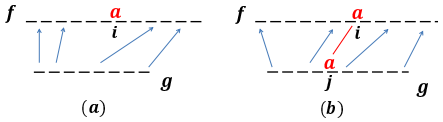

Fig. 4. Figure for the proof of Lemma 1.

$\mathbf{s}$, the number of $f|f_i = a$ is the number of empty spaces to fill in f after matching g to $f_{\mathbf{s}}$, which is $2^{n-1-|g|}$. Thus, the first term is equal to $\sum_{\mathbf{s}|i \notin \mathbf{s}} 2^{n-1-|g|} = \binom{n-1}{|g|} 2^{n-1-|g|}$. To evaluate the second term, we first note that whenever $s_j = i$, then $g_j = a$, otherwise $f_{\mathbf{s}} \neq g$. Now when $s_j = i$ and $g_j = a$, $g_{[1:j-1]}$ goes in $f_{[1:i-1]}$ and $g_{[j+1:|g|}$ goes into $f_{[i+1:n]}$, as in Fig. 4(b). As done earlier, we can now evaluate this to $2^{n-|g|} \sum_{j|g_j=1} \binom{i-1}{j-1} \binom{n-i}{|g|-j}$. □

Algorithm 1 Symbolwise MAP with 1 trace

Input: (n, Y), Output: $\hat{X}_{sm}(Y)$
for $i = 1, 2, ..., n$ **do**
  $s \leftarrow \frac{\Pr(X_i=1|Y)}{\Pr(X_i=0|Y)}$
  $X_i \leftarrow 1$ if $s \geq 1$, 0 else
return $X_{[1:n]}$

Algorithm 2 Exact Symbolwise MAP for t traces

Input: $(n, Y_1, .., Y_t)$, Output: $\hat{X}_{sm}(Y_1, ..., Y_t)$
Compute $Y_1 \uparrow Y_2 \uparrow ... \uparrow Y_t$
for $i = 1, 2, ..., n$ **do**
  $s_1, s_0 \leftarrow 0$
  for each $w \in Y_1 \uparrow Y_2$ of length$\leq n$ **do**
    $s_a \leftarrow s_a + \Big[2^{n-|w|} \langle Y_1 \uparrow ... \uparrow Y_t, w \rangle$
    $\sum_{j|w_j=a} \binom{i-1}{j-1} \binom{n-i}{|w|-j} \Big]$ for $a = 0, 1$
  $X_i \leftarrow 1$ if $s_1 \geq s_0$, 0 else
return $X_{[1:n]}$

Algorithm 3 Approximate symbolwise MAP

Input: $(n, Y_1, ..., Y_t)$, Output: $\hat{X}_{sm,approx}(Y_1, ..., Y_t)$
$s \leftarrow \prod_{j=1}^{t} \frac{\Pr(X_i=1|Y_j)}{\Pr(X_i=0|Y_j)}$
$X_i \leftarrow 1$ if $s \geq 1$, 0 else
return $X_{[1:n]}$

Corollary 1. *We can implement a polynomial time algorithm (Alg. 1) for symbolwise MAP estimation for $t = 1$ using*

$$\frac{\Pr(X_i = 1|Y)}{\Pr(X_i = 0|Y)} = \frac{\binom{n-1}{|Y|} + 2\sum_{j|Y_j=1} \binom{i-1}{j-1} \binom{n-i}{|Y|-j}}{\binom{n-1}{|Y|} + 2\sum_{j|Y_j=0} \binom{i-1}{j-1} \binom{n-i}{|Y|-j}}.$$

• $\mathbf{t > 1}$: Using similar arguments to $t = 1$, we can show the following, where $\propto$ denotes proportionality,

$$\Pr(X_i = a|Y_1 Y_2) \propto \sum_{x \in \mathcal{A}^n | x_i = a} \binom{x}{Y_1} \binom{x}{Y_2} ... \binom{x}{Y_t}. \quad (14)$$

Using Proposition 1, we see that,

$$\Pr(X_i = a|Y_1 Y_2 .. Y_t)$$
$$\propto \sum_{x|x_i=a} \sum_{w||w| \leq n} \binom{x}{w} \langle Y_1 \uparrow Y_2 \uparrow .... \uparrow Y_t, w \rangle$$
$$= \sum_{w||w| \leq n} \langle Y_1 \uparrow Y_2 \uparrow ... \uparrow Y_t, w \rangle \sum_{x|x_i=a} \binom{x}{w}. \quad (15)$$

Algorithm 2 uses (15) for the MAP criterion

$$\hat{X}_i = \arg\max_{a \in \mathcal{A}} \sum_{w||w| \leq n} \Big\{ 2^{n-|w|} \langle Y_1 \uparrow Y_2 \uparrow ... \uparrow Y_t, w \rangle$$
$$\Big(\sum_{j|w_j=a} \binom{i-1}{j-1} \binom{n-i}{|w|-j} \Big) \Big\}, \quad (16)$$

which may not be polynomial time due to the number of terms in summation of (16). We propose an alternate strategy for symbolwise estimation in Algorithm 3, where instead of computing $\arg\max_a \sum_{x \in \mathcal{A}^n | x_i = a} \binom{x}{Y_1} \binom{x}{Y_2} ... \binom{x}{Y_t}$, we equivalently compute $\arg\max_a \prod_{j=1}^{t} \Pr(X_i = a|Y_j)$. This can be computed in polynomial time from Lemma 2 and Corollary 1.

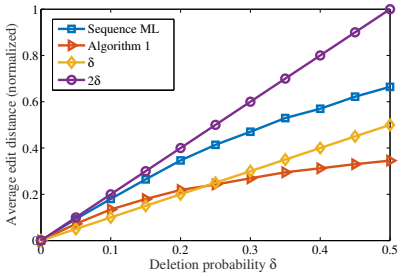

Fig. 5. $t = 1$: Alg. 1 vs. (18).

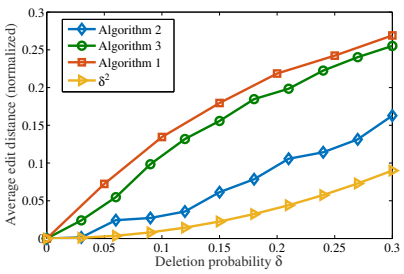

Fig. 6. $t = 2$: Algs 1, 3 and 2.

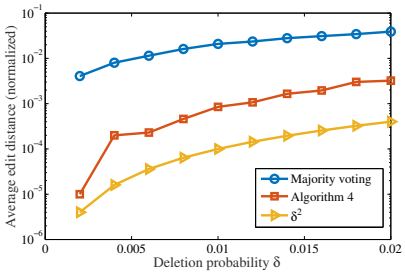

Fig. 7. Alg. 4 ($t = 2$) vs majority voting ($t = 3$).

V. Sequence Estimation

Unlike Section IV, we here assume that a-priori, we *do not* know the length of the sequence in $\mathcal{A}^*$; indeed, given a trace, the posterior probabilities for the sequence length concentrates around a range of values. Alg. 4 uses the shortest common supersequence to approximate the ML decoding of the input of the remnant channel in Section III, as well as for the overall reconstruction, given that we cannot hope to recover symbols erased from all sequences. Theorem 1 shows that the probability of the input *not* being a shortest common supersequence of two traces is smaller than the probability of the input being any given shortest common supersequence.

Algorithm 4 Shortest common supersequence reconstruction

1: Input: (Y_1, Y_2), Output: $SCS(Y_1, Y_2)$
2: Find a shortest path p in $\mathcal{G}(Y_1, Y_2)$
3: **return** $w(p)$

Theorem 1.

$$\frac{\sum\limits_{|X|>|SCS|} \Pr(X|fY_1, Y_2)}{\Pr(X = SCS(Y_1, Y_2)|Y_1, Y_2)} \leq 1.33 \left(\frac{1}{4e}\right)^{2n\epsilon}, \quad (17)$$

if $\delta \leq \left(\frac{1}{4e}\right)^{n(1+\epsilon)}$ *for any fixed* $\epsilon > 0$.

The basic idea in the proof (available in [12]) is as follows: We lower bound the denominator term in the left by 1, and upper bound the numerator by a geometric series using a variety of common algebraic inequalities.

VI. Numerical results

Single channel: We compare the symbol-wise MAP Alg. 1 to the sequence ML estimate for the single channel (with known blocklength) which is

$$\hat{X}_{ml} = \arg\max_{x \in \mathcal{A}^n} \binom{x}{Y}. \quad (18)$$

For a blocklength of 20, Fig. 5 shows that symbolwise MAP outperforms the sequence ML estimate. Note that both methods have an average edit distance less than 2δ, which is the worst case edit distance when our estimate is a random n length supersequence of Y.

Two vs. one channels: We compare the use of two channels and Algs 2-3 to using a single deletion channel and Alg. 1 for a small blocklength of 7, we used a small blocklength since simulating Alg. 3 is of exponential complexity, but we hope that the pattern follows for larger blocklengths. Fig. 6 shows that though the heuristic Alg. 3 performs better than Alg. 1, the improvement is not as significant as using the optimal MAP decoding in Alg. 2, which has significantly higher complexity. Obtaining low complexity algorithms which are closer to the optimal MAP symbol-wise decoding performance is part of on-going work. For reference we also plot δ^2, the fraction of bits deleted in both the traces (and which we can never hope to recover).

Alg. 4: We compare the performance of Alg. 4 using the edit graph vs. the majority voting reconstruction of [5] implemented for *three channels* (an odd number of channels enables to break ties during majority voting). We use a blocklength of 100 here. Fig. 7 shows that even though the majority voting uses $t = 3$ deletion channels, it can perform worse than Alg. 4 ($t = 2$). These are first results and a deeper investigation of the performance is on-going.

References

[1] W. Mao, S. N. Diggavi, and S. Kannan, "Models and information-theoretic bounds for nanopore sequencing," *2017 ISIT*, 2017.

[2] ——, "Models and information-theoretic bounds for nanopore sequencing," *CoRR*, vol. abs/1705.11154, 2017. [Online]. Available: http://arxiv.org/abs/1705.11154

[3] A. H. Laszlo et.al., "Decoding long nanopore sequencing reads of natural dna," *Nat Biotech*, pp. 829–833, 2014.

[4] V. I. Levenshtein, "Efficient reconstruction of sequences," *IEEE Transactions on Information Theory*, Jan 2001.

[5] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," in *SODA '04*, 2004, pp. 910–918.

[6] T. Holenstein, M. Mitzenmacher, R. Panigrahy, and U. Wieder, "Trace reconstruction with constant deletion probability and related results," in *ACM-SIAM SODA '08*, 2008, pp. 389–398.

[7] A. De, R. O'Donnell, and R. A. Servedio, "Optimal mean-based algorithms for trace reconstruction," in *STOC 2017*.

[8] Y. Peres and A. Zhai, "Average-case reconstruction for the deletion channel: subpolynomially many traces suffice," *CoRR*, vol. abs/1708.00854, 2017.

[9] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions and substitutions," *IEEE Transactions on Information Theory*, vol. 47, pp. 687–698, 2001.

[10] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows-wheeler transform," 2009.

[11] I. Shomorony, S. H. Kim, T. A. Courtade, and D. N. C. Tse, "Information-optimal genome assembly via sparse read-overlap graphs," *Bioinformatics*, vol. 32, no. 17, pp. i494–i502, 2016.

[12] S. Srinivasavaradhan, M. Du, S. Diggavi, and C. Fragouli, "On maximum likelihood reconstruction over multiple deletion channels," 2017. [Online]. Available: http://arni.ee.ucla.edu/v8.pdf

[13] G. Navarro, "A guided tour to approximate string matching," *ACM Comput. Surv.*, vol. 33, no. 1, pp. 31–88, Mar. 2001.

[14] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. New York, NY, USA: Cambridge University Press, 1997.

[15] M. Lothaire, *Combinatorics on Words*, ser. Cambridge Mathematical Library. Cambridge University Press, 1997.