Genomic Compression with Decoder Alignment under Single Deletion and Multiple Substitutions

Yotam Gershon Yuval Cassuto
The Viterbi Faculty of Electrical and Computer Engineering, Technion - Israel Institute of Technology
Email: {yotamgr@campus, ycassuto@ee}.technion.ac.il

***Abstract*—We address the problem of compressing genomic read data produced by modern shotgun sequencing technologies, where a reference genome, closely similar to the sequenced one, is available only at the decoder. This problem, addressed by distributed source coding techniques, requires an alignment and validation layer in the decoder. In this work, we extend a previous work, to allow a single deletion along with the previously addressed multiple substitutions. The results include a new distance for efficient alignment under deletion and substitutions, a derivation of the exact distribution of this distance on random sequences, as well as procedures to recover the read from multiple invocations of a substitutions-only decoder.**

I. Introduction

Shotgun sequencing is the process of determining the order of nucleic acids within a DNA molecule (genomic sequence), using a large set of short fragments observed randomly from the sequence. These fragments, called *reads* and represented by a string of symbols (usually A,C,G,T), are used for reconstructing the original genome, as well as for other analysis tasks. Since those reads are consumed by computationally heavy and specialized procedures, they are commonly communicated from the sequencing location (e.g. physician's office) to a central processing location (e.g. cloud genome database). Therefore, effective compression methods of the pre-assembly reads are of significant interest. Those methods are divided to reference-based and reference-free tools, differed by the use of a closely similar reference genome in the *encoding* process. Naturally, reference-based methods provide better performance in most cases, but also require significant computational and storage resources at the encoder side, which is typically limited in such resources.

In [1] we proposed a method for compressing the reads using a *reference that is available to the decoder side only*, providing reference-based performance without the cost of using the reference in the encoder. This scheme is based on a coding-theoretic solution of the Slepian-Wolf coding problem [2] using *generalized error locating* (GEL) concatenated codes. Compared to prior schemes addressing source coding with decoder side information [3]–[11], the coding scheme in [1] also addresses the problem of *aligning the compressed read* within the reference genome at the decoder side, before reconstructing the read from a matching segment. In [1] it is assumed that the differences between the reads and their corresponding segments in the reference are *substitutions only*, while practical sequencers may also introduce symbol *deletions* to the reads.

In this paper we extend the coding scheme of [1] to also handle a symbol deletion in every read, in addition to multiple substitutions. The key tool toward that is a new distance measure we propose in Section III for finding reference segments with good match to the read under the new error model. This distance, which we show to be a semi-metric, improves over prior distance measures that are either more complex to compute, or suffer from high false-alignment probabilities. Thanks to its simplicity, we are able to derive the exact distribution of the proposed distance over random sequences, using combinatorial analysis of an appropriate random-walk model. Another important ingredient of the extended scheme (Section IV) is a procedure to expand each matching candidate found by the aforementioned distance to sub-candidates that can jointly recover the read using a substitutions-only inner-code decoder. We note that supporting a single deletion in each read is without significant loss of generality, since we are free to set the read size to a value where more than one deletion is rare, and these rare failures are handled by the scheme's outer code. Moreover, it is possible to extend the proposed distance measure to multiple deletions, as well as to insertions.

II. Background and Preliminaries

A. Problem Setting and Error Model

A batch of length-n genome substrings $\{\boldsymbol{x}^{(i)}\}_{i=1}^{M}$ (called reads) is output by a sequencer, and these reads need to be communicated to a central node. The central node holds a reference genome sequence $\mathbf{Y}$ that is closely similar to the sequence from which the reads are generated. We model the similarity by an error model, which in this paper is taken to be *single deletion multiple substitutions*. That is, for each read i there exists an index k_i such that $\boldsymbol{x}^{(i)}$ is obtained from $\boldsymbol{y}^{(i)} = Y_{k_i}, \ldots, Y_{k_i+n-1}, Y_{k_i+n}$ by a single deletion and a certain number of substitutions (note that the latter has length $n+1$). Equivalently, there is an integer j_{d_i} such that

$$\boldsymbol{x}^{(i)} = \tilde{Y}_{k_i}, \ldots, \tilde{Y}_{k_i+j_{\mathsf{d}_i}-2}, \tilde{Y}_{k_i+j_{\mathsf{d}_i}}, \ldots, \tilde{Y}_{k_i+n-1}, \tilde{Y}_{k_i+n}, \quad (1)$$

and $\tilde{\mathbf{Y}}$ is the result of $\mathbf{Y}$ passing through some substitution channel. The deletion index is $k_i + j_{\mathsf{d}_i} - 1$, and a read without a deletion is simply modeled by $j_{\mathsf{d}_i} = n + 1$. This error model captures both sequencing errors and genomic diversity between the sequenced genome and the reference $\mathbf{Y}$. It is emphasized that *the encoder is unaware* of the reads' k_i indices and the reads' errors with respect to $\mathbf{Y}$, in particular whether a read contains a deletion or not.

B. Distributed Source Coding with Alignment

In this section we briefly describe the *generalized error locating* (GEL) based coding scheme proposed in [1] for the substitutions-only case.

Construction 1. Let $f_\ell(\boldsymbol{x})$ denote the sampling of ℓ predefined indices from $\boldsymbol{x}$, which will be called a *read identifier*, and let $\mathcal{I}$ denote the remaining indices. Next, let $\mathcal{C}_1, \mathcal{C}_2$ be a pair of binary linear codes with parameters $[n-\ell, k_i-\ell, d_i], i=\{1,2\}$, where $k_1 \geq k_2$. Let $\mathsf{H}_1, \mathsf{H}_2$ be parity-check matrices of these codes, respectively, such that they form a nested pair, i.e., all rows of H_1 appear in H_2 in concatenation with additional $\tau \triangleq k_1 - k_2$ rows, linearly independent on H_1, denoted by $\bar{\mathsf{H}}_2$, the *validation matrix*. Let H_c be a *complementary matrix* such that the concatenation of its rows with H_2 forms a square *full-rank* matrix H.

Finally, let $\mathcal{C}_\mathsf{o}$ be a $[M, k_\mathsf{o}, d_\mathsf{o}]$ linear code over $\mathsf{GF}(2^\nu)$, with parity-check matrix $\mathbf{H}_\mathsf{o}$, and $\nu = n - \ell - (\rho + \tau)$. For encoding, we extract $\boldsymbol{w}^{(i)} = f_\ell(\boldsymbol{x}^{(i)})$ and calculate $\boldsymbol{s}^{(i)} = \mathsf{H}_2 \left[\boldsymbol{x}_\mathcal{I}^{(i)}\right]^T$, $\boldsymbol{a}^{(i)} = \mathsf{H}_\mathsf{c} \left[\boldsymbol{x}_\mathcal{I}^{(i)}\right]^T$ for each read. We then form $\underline{\boldsymbol{a}} = \left[\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(M)}\right] \in [\mathsf{GF}(2^\nu)]^M$, and calculate $\mathbf{S} = \mathbf{H}_\mathsf{o} \underline{\boldsymbol{a}}^T$. The encoder output is $\left\{\{\boldsymbol{w}^{(i)}\}_{i=1}^M, \{\boldsymbol{s}^{(i)}\}_{i=1}^M, \mathbf{S}\right\}$, which is sent to the decoder and received without noise.

The decoding process is now briefly described. First, for every read $\boldsymbol{x}^{(i)}$, the decoder aligns the read identifier $\boldsymbol{w}^{(i)}$ over the reference to form a set of possible candidates $\mathrm{Y}^{(i)}$ by the following rule:

$$\mathrm{Y}^{(i)} = \left\{ \boldsymbol{y}^{(i,j)} \middle| \, d_H\left(f_\ell(\boldsymbol{x}^{(i)}), f_\ell(\boldsymbol{y}^{(i,j)})\right) \leq \mathsf{T} \right\}_{j=1}^{\mathsf{K}_i}, \quad (2)$$

where $d_\mathsf{H}(\cdot,\cdot)$ is the Hamming distance, T is a predefined threshold, and $\left\{\boldsymbol{y}^{(i,j)} = \left[Y_{k_i^{(j)}}, \ldots, Y_{k_i^{(j)}+n-1}\right]\right\}$ is a substring of $\mathbf{Y}$ *closely matching* to the read. Next, every candidate is decoded with respect to H_1 within the coset of syndrome $\boldsymbol{s}_1^{(i)}$. The result $\boldsymbol{v}$ is *validated* using $\mathcal{C}_2$ by testing whether $\bar{\mathsf{H}}_2 \boldsymbol{v}^T = \boldsymbol{s}_2^{(i)}$. This validation is a key stage in the coding scheme, allowing the disqualification of *improper alignments*, i.e., candidates erroneously aligned based on a randomly matching read identifier. If exactly one candidate from $\mathrm{Y}^{(i)}$ is decoded to a word $\boldsymbol{v}$ being validated, $\boldsymbol{b}^{(i)} = \mathsf{H}_\mathsf{c} \boldsymbol{v}^T$ is calculated. Otherwise, an erasure is set: $\boldsymbol{b}^{(i)} = \otimes$. Next, $\underline{\boldsymbol{b}} = \left[\boldsymbol{b}^{(i)}, \ldots, \boldsymbol{b}^{(M)}\right]$ is decoded with respect to $\mathbf{H}_\mathsf{o}$ within the coset of syndrome $\mathbf{S}$ to form $\hat{\underline{\boldsymbol{a}}}$. This is called the *outer decoding* over an *outer channel*. Finally, each read is reconstructed from $\boldsymbol{w}^{(i)}$ and $\hat{\boldsymbol{x}}_\mathcal{I}^{(i)} = \mathcal{F}_\mathsf{H}([\boldsymbol{s}^{(i)}, \hat{\boldsymbol{a}}^{(i)}])$, where $\mathcal{F}_\mathsf{H}(\boldsymbol{u})$ is the linear mapping of $\boldsymbol{u}$ to the single codeword of syndrome $\boldsymbol{u}$ in the code defined by H.

III. A Distance Measure for Alignment with a Deletion

A. Motivation

In order to extend the coding scheme to deal with deletion errors, it is necessary to establish an efficient way to perform alignment under such errors, along with substitution errors. With substitutions only, an offset in the reference $\mathbf{Y}$ is considered a good alignment candidate if its Hamming distance to the read is small (see (2)). Now we need an alternative distance measure that allows a single deletion anywhere in the offset's subsequence before evaluating its Hamming distance to the read. We note that in the actual scheme this alignment is performed using only the identifier of the read (see the use of $f_\ell(\cdot)$ in (2)), but for simplicity in this section we assume the full read is aligned.

B. Existing Distance Measures

An immediate candidate for such a measure is the Levenshtein distance [12], counting the minimal number of edits (deletions, insertions and substitutions) required to obtain one word from another. Nevertheless, this measure suffers from two main issues in our case: 1) its complex calculation by dynamic-programming algorithms makes it impractical to evaluate each read along every possible offset position in the reference, 2) allowing unrestricted error patterns, involving any number of deletions, insertions and substitutions, introduces unfitting alignment candidates.

Another possible measure is the shifted Hamming distance [13], which matches each read index with r adjacent indices in the subsequence at the offset considered for alignment. For our purposes, since only deletions (and not insertions) are relevant, we use only adjacent indices to the right of the original index. This can be formalized by

$$\forall \boldsymbol{x} \in \boldsymbol{\Sigma}^n, \boldsymbol{y} \in \boldsymbol{\Sigma}^{n+r} : d_{\mathsf{SH}}(\boldsymbol{x}, \boldsymbol{y}) \triangleq \sum_{i=1}^{n} \bigwedge_{j=0}^{r} x_i \oplus y_{i+j}, \quad (3)$$

where $\wedge$ denotes a logical 'AND' operation, and $\boldsymbol{\Sigma}^m$ denotes a word of length m from alphabet $\boldsymbol{\Sigma}$, and the binary operator $\oplus$ returns 0 for equal symbols and 1 otherwise. The main issue with this measure is that by allowing independent shifts between indices, ignoring the special shift structure of index deletion, even random unrelated sequences may be declared close, increasing the number of improper alignments.

C. Preliminaries

We first give two preliminary definitions that will help defining the new distance measure $\delta_{\mathsf{s.c}}(\boldsymbol{x}, \boldsymbol{y})$ in the next sub-section. At the end of this section, the distance measure is extended to a formal *semi-metric* $\mathsf{d}_{\mathsf{s.c}}(\cdot,\cdot)$ satisfying the symmetry and triangle-inequality metric properties, in addition to a natural generalization of the third metric property: the identity of indiscernibles. From this point, we assume $\boldsymbol{\Sigma} = \{0,1\}$ for simplicity, but every result can be extended to any alphabet size q.

Definition 1. ***(Cumulative Hamming distance)*** *For $\boldsymbol{x} \in \boldsymbol{\Sigma}^n$, $\boldsymbol{y} \in \boldsymbol{\Sigma}^{n+r}$, define, for every $0 \leq j \leq r$ and every $0 \leq t \leq n$,*

$$\phi_j(\boldsymbol{x}, \boldsymbol{y}; t) \triangleq \sum_{i=1}^{t} x_i \oplus y_{i+j},$$

where $\oplus$ denotes an addition over $\mathsf{GF}(2)$. When clear from the context, we will denote $\phi_j(\boldsymbol{x}, \boldsymbol{y}; t) = \phi_j(t)$.

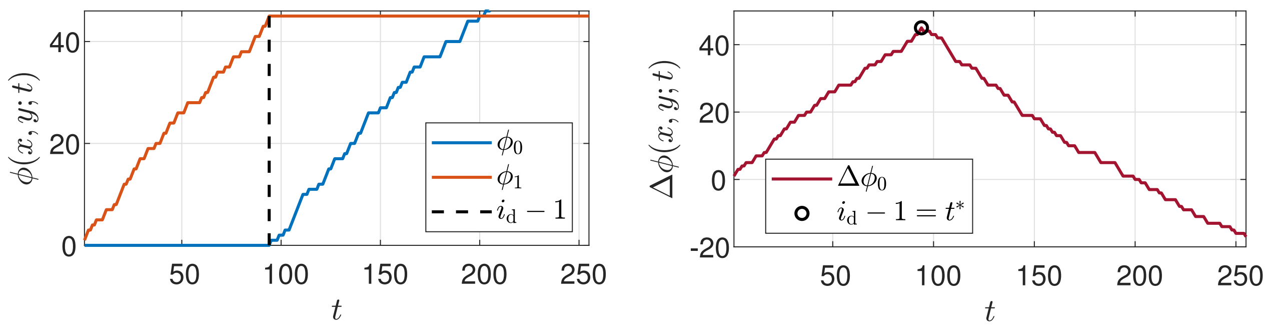

Fig. 1: Illustration of the shift compensating distance components, with no substitutions: cumulative distances (left), and their difference (right), with the deletion location minus 1 attaining the maximal difference.

Definition 2. *Let $\boldsymbol{x} \in \boldsymbol{\Sigma}^n$, $\boldsymbol{y} \in \boldsymbol{\Sigma}^{n+r}$. Define, for every $0 \leq j \leq r-1$ and every $0 \leq t \leq n$,*

$$\begin{aligned}\Delta\phi_j(\boldsymbol{x},\boldsymbol{y};t) &\triangleq \phi_{j+1}(\boldsymbol{x},\boldsymbol{y};t) - \phi_j(\boldsymbol{x},\boldsymbol{y};t)\\ &= \sum_{i=1}^{t}[x_i \oplus y_{i+j+1}] - [x_i \oplus y_{i+j}],\end{aligned}$$

where by definition $\Delta\phi_j(\boldsymbol{x},\boldsymbol{y};0)=0$. Again, when clear from the context, we will denote $\Delta\phi_j(\boldsymbol{x},\boldsymbol{y};t)=\Delta\phi_j(t)$.

D. Single-Deletion Compensating Distance Measure

We can now introduce the desired measure for underlying Hamming distance between two words, while compensating for a block shift caused by a single deletion occurred in one of them before being transmitted through a substitution channel.

Definition 3. *Let $\boldsymbol{x} \in \boldsymbol{\Sigma}^n, \boldsymbol{y} \in \boldsymbol{\Sigma}^{n+1}$, and let*

$$t^* \triangleq \arg\max_{0\leq t\leq n}\{\Delta\phi_0(\boldsymbol{x},\boldsymbol{y};t)\}. \tag{4}$$

*Then, we define the **shift-compensating distance** by*

$$\begin{aligned}\delta_{\mathsf{s.c}}(\boldsymbol{x},\boldsymbol{y}) &\triangleq \phi_1(\boldsymbol{x},\boldsymbol{y};n) - \Delta\phi_0(\boldsymbol{x},\boldsymbol{y};t^*)\\ &= \min_{0\leq t\leq n}\{\phi_1(\boldsymbol{x},\boldsymbol{y};n) - \Delta\phi_0(\boldsymbol{x},\boldsymbol{y};t)\}.\end{aligned}$$

$\delta_{\mathsf{s.c}}(\boldsymbol{x},\boldsymbol{y})$ measures the "split" distance where the non-shifted $\boldsymbol{y}$ is used until index t^*, and the shifted $\boldsymbol{y}$ thereafter; t^* is the index that maximizes the gap between the shifted and non-shifted cumulative distances, indicating t^*+1 is a likely deletion position. This measure is illustrated in Fig. 1, and its properties are formalized in the next lemma (proof omitted).

Lemma 4. *Let $\boldsymbol{x} \in \boldsymbol{\Sigma}^n, \boldsymbol{y} \in \boldsymbol{\Sigma}^{n+1}$. Let $\boldsymbol{y}^{[k]}$ denote the word obtained from $\boldsymbol{y}$ by a deletion in index k. Then,*

$$\delta_{\mathsf{s.c}}(\boldsymbol{x},\boldsymbol{y}) = \min_{1\leq k\leq n+1}\left\{d_{\mathsf{H}}(\boldsymbol{x},\boldsymbol{y}^{[k]})\right\}, \tag{5}$$

and $t^* = \arg\min_{1\leq k\leq n+1}\left\{d_{\mathsf{H}}(\boldsymbol{x},\boldsymbol{y}^{[k]})\right\}-1$, where t^* is as defined in (4).

The fact that $\delta_{\mathsf{s.c}}$ in (5) is equal to the minimum substitution distance over all possible deletion indices shows that it is the natural distance measure when aligning with a single deletion, and thus will perform better than d_{SH} in (3). Importantly, it still has a low (linear) calculation complexity, making it more practical than other alternatives such as the Levenshtein distance.

We end the presentation of this distance measure by defining a variation of it that we can prove (omitted) to be a semi-metric.

Definition 5. *Let $\boldsymbol{x},\boldsymbol{y} \in \boldsymbol{\Sigma}_n^{n+1}$. The **shift-compensating semi-metric** is defined by*

$$\mathsf{d}_{\mathsf{s.c}}(\boldsymbol{x},\boldsymbol{y}) \triangleq \begin{cases} d_{\mathsf{H}}(\boldsymbol{x},\boldsymbol{y}) & ,|\boldsymbol{x}|=|\boldsymbol{y}| \\ \delta_{\mathsf{s.c}}(\boldsymbol{x},\boldsymbol{y}) & ,|\boldsymbol{x}|=|\boldsymbol{y}|-1 \\ \delta_{\mathsf{s.c}}(\boldsymbol{y},\boldsymbol{x}) & ,|\boldsymbol{x}|=|\boldsymbol{y}|+1 \end{cases}.$$

We can prove that $\mathsf{d}_{\mathsf{s.c}}(\cdot,\cdot)$ is a semi-metric satisfying a generalized identity of indiscernibles by which a length-n $\boldsymbol{x}$ is at distance 0 from all length-$(n+1)$ $\boldsymbol{y}$ vectors in its radius-1 insertion ball. This captures well the fact that multiple length-$(n+1)$ sequences are 1 deletion and 0 substitutions away from a length-n alignment target.

E. Analysis of Alignment Performance

In this sub-section we assume that the reference sequence $\mathbf{Y}$ (defined in Section II-A) is a random binary sequence in which each symbol is drawn i.i.d from the Bernoulli$(1/2)$ distribution. We also assume that $\tilde{\mathbf{Y}}$ is obtained from $\mathbf{Y}$ by passing each symbol through a binary symmetric channel with parameter p. Throughout this sub-section, we seek to align a length-n vector $\boldsymbol{x}$ with the reference $\mathbf{Y}$:

Definition 6. *Let $\boldsymbol{x} \in \boldsymbol{\Sigma}^n$ and let $\boldsymbol{y} \in \boldsymbol{\Sigma}^{n+1}$ taken from some offset in $\mathbf{Y}$. We say that $\boldsymbol{y}$* **matches** *$\boldsymbol{x}$ if $\delta_{\mathsf{s.c}}(\boldsymbol{x},\boldsymbol{y}) \leq \mathsf{T}_{\mathsf{d}}$, for some integer distance threshold T_{d}.*

Recall from (1) that in our problem $\boldsymbol{x}=\tilde{\boldsymbol{y}}^{[k]}$, i.e., $\boldsymbol{x}$ is obtained by taking a consecutive subsequence $\tilde{\boldsymbol{y}}$ of $\tilde{\mathbf{Y}}$ and deleting from it the k-th symbol. We want to evaluate the probability that $\boldsymbol{y}$ matches $\boldsymbol{x}$, and we are interested in two cases: when $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ have the same offset in $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$, respectively *(proper alignment)*, and when the offsets of $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ are different *(improper alignment)*.

We start from the simpler case of proper alignment. $\mathsf{F}_{\mathsf{b}}(n,p,\mathsf{T})$ denotes the cumulative distribution function (CDF)

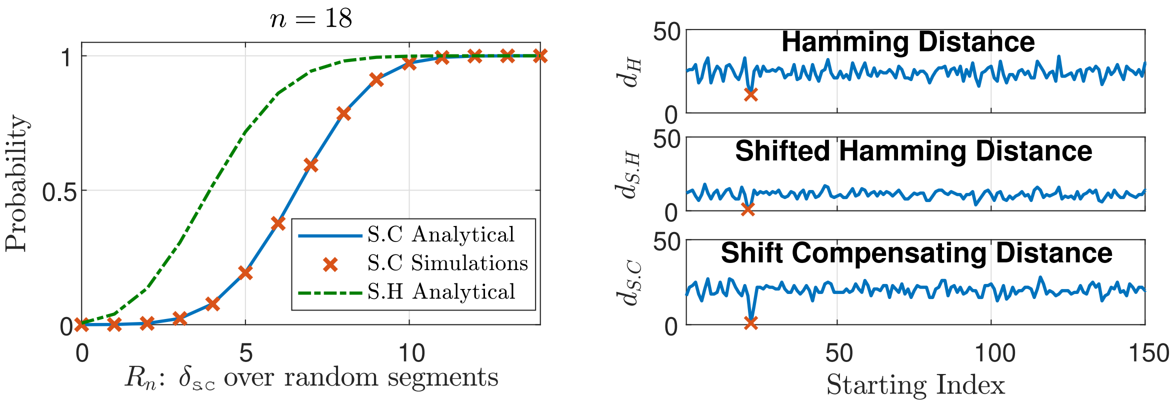

Fig. 2: **Left:** CDF of the shift-compensating distance (S.C) in comparison to the shifted Hamming distance (S.H). **Right:** Comparing the three distances when a read is aligned over a noisy version of the genome.

of a binomial random variable with parameters (n, p), evaluated at the value of T.

Proposition 7. *The probability $\mathcal{P}_{\mathsf{a.s}}$ that the proper-alignment $\boldsymbol{y}$ matches $\boldsymbol{x}$ satisfies $\mathcal{P}_{\mathsf{a.s}} \geq \mathsf{F}_{\mathsf{b}}(n, p, \mathsf{T}_{\mathsf{d}})$.*

We note that equality is obtained in the case of substitutions only and replacing $\delta_{\mathsf{s.c}}(\boldsymbol{x}, \boldsymbol{y})$ by $d_{\mathsf{H}}(\boldsymbol{x}, \boldsymbol{y})$; proving this with inequality in Proposition 7 is immediate from (5), showing that $d_{\mathsf{H}}(\boldsymbol{x}, \tilde{\boldsymbol{y}}^{[k]})$ is an upper bound on $\delta_{\mathsf{s.c}}(\boldsymbol{x}, \boldsymbol{y})$, thus exceeding the threshold T_{d} on the latter implies more than T_{d} substitutions in $\tilde{\boldsymbol{y}}^{[k]}$. For analyzing improper alignments, we study the probability distribution of the shift-compensated distance when evaluated on a *random* $\boldsymbol{y}$ word unrelated to $\boldsymbol{x}$. We first observe that for two independent random words $\boldsymbol{x} \in \boldsymbol{\Sigma}^n, \boldsymbol{y} \in \boldsymbol{\Sigma}^{n+1}$, each chosen uniformly from the entire space, we can write $\Delta\phi_0(\boldsymbol{x}, \boldsymbol{y}; t) = \sum_{i=0}^{t} \Delta_i, 1 \leq t \leq n$, where $\Delta_0 = 0$, and $\{\Delta_i\}_{i=1}^{t}$ are independent random variables with support $\{-1, 0, 1\}$ and probabilities $\{0.25, 0.5, 0.25\}$, respectively. This sum forms a *symmetric random walk with null steps*, which provides the framework for the next theorem.

Theorem 8. *Let us denote by R_n the random variable of $\delta_{\mathsf{s.c}}(\boldsymbol{x}, \boldsymbol{y})$ for randomly chosen $\boldsymbol{x} \in \boldsymbol{\Sigma}^n, \boldsymbol{y} \in \boldsymbol{\Sigma}^{n+1}$. Then,*

$$P(R_n = r) =$$

$$\frac{1}{4^n} \sum_{m=0}^{n-r} \sum_{t=m}^{n} \sum_{k=0}^{n-m} \sum_{w=0}^{k} \sum_{l=0}^{t-1} A_1(t, w, m, l) \sum_{v=0}^{n-t-(k-w)} A_2(v, t, k-w, r-l),$$

where we defined

$$A_1(t, w, m, l) \triangleq \frac{m}{t} \cdot \binom{t}{\alpha, \alpha+m, \beta, w-\beta},$$

$$A_2(v, t, k-w, r-l) = \frac{v+1}{\gamma+v+1} \binom{n-t}{\gamma, \gamma+v, \eta, k-w-\eta},$$

for $\alpha \triangleq (t-w-m)/2, \beta \triangleq l-\alpha, \gamma \triangleq (n-t-(k-w)-v)/2, \eta \triangleq r-l-\gamma$, and $\binom{z}{u_1, u_2, \ldots, u_m} \triangleq \frac{z!}{u_1! u_2! \ldots u_m!}$, the multinomial coefficient.

Proof (idea): To count the number of sequence pairs that have $\delta_{\mathsf{s.c}}(\boldsymbol{x}, \boldsymbol{y}) = r$, we classify the pairs according to several variables appearing as summation indices. The primary variables are t that is the value t^* calculated in (4), and m that is $\Delta\phi_0(\boldsymbol{x}, \boldsymbol{y}; t^*)$. Given t, m, we count the number of random walks of Δ_i that attain a global maximum of m at time t, and for each such walk expand the number of sequence pairs that have distance r. The variables k and w count the number of zeros of the random walk in total and until time t, respectively. The remaining indices are weight variables: l for the subsequence $x_i \oplus y_i$ until time t, and v for the subsequence $x_i \oplus y_{i+1}$ thereafter. ■

Proposition 9. *The probability $\mathcal{P}_{\mathsf{f.a.s}}$ that the improper-alignment $\boldsymbol{y}$ matches $\boldsymbol{x}$ satisfies $\mathcal{P}_{\mathsf{f.a.s}} \approx P(R_n \leq \mathsf{T}_{\mathsf{d}})$.*

The left side of Fig. 2 shows an example for the distribution derived in Theorem 8 (blue line), validated by an empirical calculation (red crosses). The large gap in comparison to the shifted-Hamming distribution (green dashed line) demonstrates the advantage of the proposed metric in rejecting improper alignments. The right side of Fig. 2 illustrates the distances obtained in aligning a read of length $n = 50$ over a noisy reference containing 0.01 substitutions rate, using Hamming, shifted Hamming and shift-compensating distances, for all possible starting indexes. The correct starting index is marked with a red cross. It can be seen that the shift-compensating distance gives the largest margins between correct and incorrect alignment indices.

IV. Extending the Coding Scheme to Deletions

In this section, we describe the modifications needed to extend the coding scheme of Section II-B to support reads that may contain a deletion. The main tool in this extension is the shift-compensating distance of Definition 3 that replaces the standard Hamming distance (see (2)) used in the case of substitutions only.

A. Extending $\delta_{\mathsf{s.c}}$ to Non-Consecutive Read Identifiers

In the coding scheme, we align to $\mathbf{Y}$ a partial read identifier of $\boldsymbol{x}$, and not $\boldsymbol{x}$ itself as assumed in the previous section. It is straightforward to extend $\delta_{\mathsf{s.c}}$ to the case where the contents of $\boldsymbol{x}$ are available only at a set $1 \leq i_1 < i_2 < \cdots < i_{\ell_{\mathsf{d}}} \leq n$

of non-consecutive indices. We first modify the cumulative Hamming distance:

$$\phi_j^{'}(\boldsymbol{x}, \boldsymbol{y}; t) \triangleq \sum_{k=1}^{t} x_{i_k} \oplus y_{i_k+j}, \ \text{ for } \ 1 \leq t \leq \ell_{\mathsf{d}},$$

and then define $\delta'_{\mathsf{s.c}}$ as in Definition 3 using $\phi_j^{'}$. We can now align any read identifier $\boldsymbol{w}^{(i)}$ to the reference $\mathbf{Y}$, to find matches as defined in Definition 6. Each match yields a candidate $\boldsymbol{y}^{(i,j)}$, which is now a substring of $\mathbf{Y}$ of length $n+1$, and all the matches form the set $\mathrm{Y}^{(i)}$.

B. Generating Sub-Candidates for Each Alignment Candidate

Matching a subsequence $\boldsymbol{y}'$ of $\mathbf{Y}$ to $\boldsymbol{w}^{(i)}$ and placing it in $\mathrm{Y}^{(i)}$ is the first step to recover $\boldsymbol{x}^{(i)}$ at the decoder. The next step is to use the syndrome $\boldsymbol{s}^{(i)}$ to correct the substitution errors between $\boldsymbol{x}^{(i)}$ and a vector obtained from $\boldsymbol{y}^{(i,j)} \in \mathrm{Y}^{(i)}$ by a deletion in one of its indices. Toward that, for each $\boldsymbol{y}^{(i,j)}$ the extended scheme generates a list of sub-candidates $\{\boldsymbol{y}_s^{(i,j)}\}$, according to the following procedure. Define χ to be a global integer tolerance parameter, and denote by $t \in \{0, \dots, \ell_{\mathsf{d}}\}$ the value of t^* that $\boldsymbol{y}^{(i,j)}$ yielded in Definition 3. Then the set $\{\boldsymbol{y}_s^{(i,j)}\}$ is defined by all deletion indices τ that satisfy the following:

$$\tau \in \mathcal{I}_{\chi}(t) \triangleq \begin{cases} [i_{t-\chi}, i_{t+\chi+1}] & , \chi < t < \ell_{\mathsf{d}} - \chi - 1 \\ [1, i_{t+\chi+1}] & , 0 \leq t \leq \chi \\ [i_{t-\chi}, n+1] & , \ell_{\mathsf{d}} - \chi - 1 \leq t \leq \ell_{\mathsf{d}} \end{cases}, \quad (6)$$

that is, all deletions within $2\chi+1$ intervals of the read identifier's indices around t, with exceptions at the extremal indices. For $\chi = 0$, i.e. no tolerance, we have $\mathcal{I}_0(t) = [i_t, i_{t+1}]$, whereas for $\chi \geq \max\{t, \ell_{\mathsf{d}} - t\}$ we have $\mathcal{I}_{\chi}(t) = [1, n+1]$, i.e., $\{\boldsymbol{y}_s^{(i,j)}\} = \mathrm{D}_1(\boldsymbol{y}^{(i,j)})$, where $\mathrm{D}_1(\cdot)$ denotes the single-deletion ball. It is motivated to use $\chi > 0$ because the true deletion index in $\boldsymbol{y}^{(i,j)}$ may fall outside its estimated interval defined by t. The value of χ controls the number of vectors qualifying to recover $\boldsymbol{x}^{(i)}$ by inner-code decoding, and it is set[1] to best balance successful recovery from the proper alignments with effective rejection of false candidates.

C. Inner-code Decoding with Multiple Sub-Candidates

Recall from Section II-B that in the substitutions-only scheme each candidate $\boldsymbol{y}^{(i,j)}$ is passed to inner-code decoding and validation. Now with deletions, we need to decode and validate a *set of sub-candidates* $\{\boldsymbol{y}_s^{(i,j)}\}$. To see how this should be done, we note the following observation.

Observation 10. *Let* $\boldsymbol{z}^{[k]} \in \mathrm{D}_1(\boldsymbol{z})$, *and let* $\boldsymbol{x} = \boldsymbol{z}^{[i_{\mathsf{d}}]}$. *Then, for* $e_i \triangleq \boldsymbol{z}_i \oplus \boldsymbol{z}_{i+1}$,

$$\boldsymbol{x}_i \oplus \boldsymbol{z}_i^{[k]} = \begin{cases} 0 & , i < \min(k, i_{\mathsf{d}}) \text{ or } i \geq \max(k, i_{\mathsf{d}}) \\ e_i & , \text{otherwise} \end{cases}.$$

Observation 10 means that if i_{d} is the actual deletion index and k is the one chosen for some sub-candidate, then this mismatch may introduce errors only between those two indices. Hence it is likely that for the proper alignment of $\boldsymbol{x}^{(i)}$ multiple sub-candidates with deletion indices close to i_{d} will correctly decode to $\boldsymbol{x}^{(i)}$. This motivates the following treatment of $\{\boldsymbol{y}_s^{(i,j)}\}$ in the modified scheme. For every candidate $\boldsymbol{y}^{(i,j)} \in \mathrm{Y}^{(i)}$: (1) Decode and validate every sub-candidate $\boldsymbol{u} \in \{\boldsymbol{y}_s^{(i,j)}\}$, (2) Store any $\boldsymbol{v}$ that was successfully decoded and validated, and the number of its appearances $A(\boldsymbol{v})$ along the decoding instances in $\{\boldsymbol{y}_s^{(i,j)}\}$, (3) Apply a majority rule on the set:

$$\mathsf{Maj}(\mathcal{V}) = \begin{cases} \boldsymbol{v}^* & , \forall \boldsymbol{v} \in \mathcal{V} \setminus \{\boldsymbol{v}^*\} : A(\boldsymbol{v}^*) > A(\boldsymbol{v}) \\ \emptyset & , \text{there exist no such } \boldsymbol{v}^* \end{cases}. \quad (7)$$

From this point, the word $\boldsymbol{v}^*$ takes the place of $\boldsymbol{v}$ as defined in the decoding of Construction 1, and the rest of the decoding process is unaltered. Note that the encoding process is also unchanged. The modified decoding process is summarized in Algorithm 1, where we denote $\mathcal{D}_1(\boldsymbol{z}, \boldsymbol{s})$ as the result of decoding the word $\boldsymbol{z}$ with respect to H_1 within the coset of syndrome $\boldsymbol{s}$, and similarly for $\mathcal{D}_{\mathsf{o}}(\boldsymbol{a}, \mathbf{S})$, with a word $\boldsymbol{a}$ decoded with respect to $\mathbf{H}_{\mathsf{o}}$ to a syndrome $\mathbf{S}$. The derivation of outer channel probabilities and optimal outer redundancy are omitted, and are analyzed similarly to the analysis in [1].

Algorithm 1: Decoding Construction 1 with Deletions

```
Input: $\mathcal{E}\left(\{\boldsymbol{x}^{(i)}\}_{i=1}^{M}\right), \mathbf{Y}, \mathsf{H}_1, \bar{\mathsf{H}}_2, \mathsf{H}_{\mathsf{c}}, \mathbf{H}_{\mathsf{o}}$
for $1 \leq i \leq M$ do
    Align $\boldsymbol{w}^{(i)}$ over $\mathbf{Y}$, and form $\mathrm{Y}^{(i)}$
    Set 'found' $\leftarrow 0$
    for $1 \leq j \leq |\mathrm{Y}^{(i)}|$ do // Inner Decoding
        Set $\mathcal{V} = \emptyset$
        for every $\boldsymbol{u} \in \{\boldsymbol{y}_s^{(i,j)}\}$ do
            Decode $\boldsymbol{v} = \mathcal{D}_1(\boldsymbol{u}_{\mathcal{I}}, \boldsymbol{s}_1^{(i)})$
            Calculate $\hat{\boldsymbol{s}}_2^{(i)} = \bar{\mathsf{H}}_2 \boldsymbol{v}^T$
            if $\hat{\boldsymbol{s}}_2^{(i)} = \boldsymbol{s}_2^{(i)}$ then // Validation
                $\mathcal{V} \leftarrow \mathcal{V} \cup \{\boldsymbol{v}\}, \ A(\boldsymbol{v}) = A(\boldsymbol{v}) + 1$
        $\boldsymbol{v}^* = \mathsf{Maj}(\mathcal{V})$ (Eq. 7)
        if $\boldsymbol{v}^* \neq \emptyset$ then // Appropriate candidate
            if 'found' $= 0$ then
                Calculate $\boldsymbol{b}^{(i)} = \mathsf{H}_{\mathsf{c}}(\boldsymbol{v}^*)^T$
                Set 'found' $\leftarrow 1$
            else // More Than One Candidate
                Set $\boldsymbol{b}^{(i)} = \bigotimes$, break
    if 'found' $= 0$ then // No Candidates
        Set $\boldsymbol{b}^{(i)} = \bigotimes$
// Outer Decoding
Decode $\underline{\hat{\boldsymbol{a}}} = \mathcal{D}_{\mathsf{o}}(\underline{\boldsymbol{b}}, \mathbf{S})$, where $\underline{\boldsymbol{b}} = [\boldsymbol{b}^{(1)}, \dots, \boldsymbol{b}^{(M)}]$
for $1 \leq i \leq M$ do // Inverse Mapping
    Map $\hat{\boldsymbol{x}}_{\mathcal{I}}^{(i)} = \mathcal{F}_{\mathsf{H}}([\boldsymbol{s}^{(i)}, \hat{\boldsymbol{a}}^{(i)}])$
    Reconstruct $\hat{\boldsymbol{x}}^{(i)}$ from $\hat{\boldsymbol{x}}_{\mathcal{I}}^{(i)}, \boldsymbol{w}^{(i)}$
Output: $\{\hat{\boldsymbol{x}}^{(i)}\}_{i=1}^{M}$
```

V. Acknowledgement

This work was supported in part by the Israel Science Foundation. The authors thank Jacob Ziv for valuable discussions.

[1] based on the other scheme parameters and the error statistics.

REFERENCES

[1] Y. Gershon and Y. Cassuto, “Distributed Source Coding of Fragmented Genomic Sequencing Data,” *IEEE International Symposium on Information Theory (ISIT)*, 2021.

[2] D. Slepian and J. K. Wolf, “Noiseless Coding of Correlated Information Sources,” *IEEE Trans. on Information Theory, Vol. IT-19, No. 4*, 1973.

[3] S. S. Pradhan and K. Ramchandran, “Distributed Source Coding Using Syndromes (DISCUS): Design and Construction,” *IEEE Trans. on Information Theory, Vol. 49, No. 3*, 2003.

[4] T. Uyematsu, “An Algebraic Construction of Codes for Slepian-Wolf Source Networks,” *IEEE Trans. on Information Theory, Vol. 47, No. 7*, 2001.

[5] A. Orlitsky and K. Viswanathan, “One-Way Communication and Error-Correcting Codes,” *IEEE Trans. on Information Theory, Vol. 49, No. 7*, 2003.

[6] Y. Minsky, A. Trachtenberg, and R. Zippel, “Set Reconciliation with Nearly Optimal Communication Complexity,” *IEEE Trans. on Information Theory, Vol. 49, No. 9*, 2003.

[7] A. Aaron and B. Girod, “Compression with Side Information Using Turbo Codes,” *Proceedings of IEEE Data Compression Conference*, 2002.

[8] M. Sartipi and F. Fekri, “Distributed Source Coding in Wireless Sensor Networks using LDPC coding: The entire Slepian-Wolf Rate Region,” *IEEE Wireless Communications and Networking Conference*, 2005.

[9] Y. Cassuto and J. Ziv, “Efficient Compression of Long Arbitrary Sequences with No Reference at the Encoder,” *IEEE Trans. on Information Theory, Vol. 67, No. 1*, 2021.

[10] S. Wang, X. Jiang, F. Chen, L. Cui, and S. Cheng, “Streamlined Genome Sequence Compression using Distributed Source Coding,” *Cancer Informatics. Supplementary Issue: Computational Advances in Cancer Informatics*, 2014.

[11] J. J. Shao, “Genome Sequence Compression Algorithm Based on the Distributed Source Coding,” *6th International Conference on Machinery, Materials, Environment, Biotechnology and Computer (MMEBC)*, 2016.

[12] V. I. Levenshtein, “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals,” *Soviet Physics Doklady*, 1966.

[13] H. Xin, J. Greth, J. Emmons, G. Pekhimenko, C. Kingsford, C. Alkan, and O. Mutlu, “Shifted Hamming Distance: A Fast and Accurate SIMD-Friendly Filter to Accelerate Alignment Verification in Read Mapping,” *Bioinformatics, 31(10)*, 2015.