Efficient Distributed Source Coding of Fragmented Genomic Sequencing Data

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

***Abstract*—In this paper we present a new compression scheme for genomic read data produced by modern sequencing technologies. In this setting, a reference genome similar to the one being sequenced is available only at the decoder, while the starting index of each read in this reference in unknown. The proposed scheme significantly reduces the encoding complexity relative to known reference-based compression schemes. The results include a code construction based on generalized concatenation coset codes, analysis of the decoding failure probability, and optimization of the scheme parameters for minimal compression rate.**

I. Introduction

Genomic sequencing is a process in which the order of nucleic acids within a DNA molecule (genomic sequence) is analyzed. In most modern sequencing technologies, a large set of short sequence fragments, called *reads*, is produced and represented by a string of characters (usually *A,C,G,T*). In this method, called *shotgun sequencing* [1], each read's location within the sequence is generally unknown. Furthermore, the sequencing machine introduces mutation errors into the reads. Therefore, the sequence assembly from the reads requires a large number of reads and high computational effort. Effective compression of pre-assembly reads data is therefore an essential problem. A large number of read-compression methods are available [2]–[4], partitioned into two main categories: reference-free and reference-based tools, differing by whether a closely similar reference genome is shared by the encoder and decoder. In some applications, genome reads will be produced at an edge node, e.g. a physician's office, and then sent for processing to a central node, e.g. a cloud database. In such scenarios, the cost of known reference-based methods may be too high for the edge node's limited resources, due to the need to store long references and perform the reads' alignment on them. Alternatively, in this paper we propose a compression scheme in which the encoder needs to neither store nor process any reference, while benefiting from a reference available at the central node decoding the reads.

Compression with reference available only at the decoder is an instance of the well known Slepian-Wolf (SW) coding problem [5]. Several works have addressed the problem with explicit constructive codes [6]–[12], some specifically for genomic data [13], [14], but focused on either full-block or streamlined data, and not fragmented reads. When compressing fragmented reads, the known capability of recovering a read from a similar reference is not sufficient, and additional information is needed for aligning the read within the full reference. The construction proposed in this paper offers both capabilities, using the framework of generalized error-locating (GEL) [15] *coset codes*. GEL coset codes were used in [12] for full-block compression, and the novelty of this present work is in using the GEL's hierarchy of inner codes to combine all the information for read recovery into the same codeword: one layer for alignment, one for similarity reconstruction and one for alignment validation. In the fourth layer, a batch of reads is encoded with an outer code to provide extremely low failure probability. In addition to proposing the code construction in Section III, in Section IV we analyze the success probability of the code, taking into account the effect of read misalignments within the reference. In Section V, we seek the optimization of the various code parameters to minimize the compression rate for a specified success probability. The similarity measure considered in this paper is the rate of *substitutions*, which are the most dominant mutations observed in common sequencing technologies. Extending the scheme to deal with insertions and deletions is left as an interesting future work.

II. Problem Formulation

A. Problem Setting

A genome data being sequenced in an edge node is represented by a string $\mathbf{X} \in \mathcal{A}^L$. A closely similar reference of this data, represented by $\mathbf{Y}$, is stored in a central node, and is unavailable at the edge node. The differences between $\mathbf{X}$ and $\mathbf{Y}$ are assumed to be caused by genetic diversity [16]. A sequencing machine at the edge node is generating a set of M *reads* from $\mathbf{X}$, denoted by $\{\boldsymbol{x}^{(i)}\}_{i=1}^{M}$. Each read $\boldsymbol{x}^{(i)} \in \mathcal{A}^n$ is an *approximate substring* of $\mathbf{X}$, taken from an unknown random location within $\mathbf{X}$, and introduced with *sequencing errors*. Our goal is to encode the reads data $\{\boldsymbol{x}^{(i)}\}_{i=1}^{M}$ to a minimal size, such that the reads can be recovered from the similar reference $\mathbf{Y}$ without loss. The encoder output is communicated to the central node without errors.

B. Error Channel Model

In this work, we assume that both genomic diversity and sequencing errors are modeled by substitutions only, without insertions and deletions (indel errors), but supporting indels is possible with modification of the alignment procedure at the central node. A q-ary, length-L *substitution channel* with substitution probability p_1, between input and output strings $\mathbf{X} = X_1, \ldots, X_L$ and $\mathbf{Y} = Y_1, \ldots, Y_L$, is defined as a memoryless q-ary symmetric channel, that is:

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{L} P(Y_j|X_j),\ P(Y_j|X_j) = \begin{cases} 1-p_1 & Y_j = X_j \\ p_1/(q-1) & Y_j \neq X_j \end{cases}$$

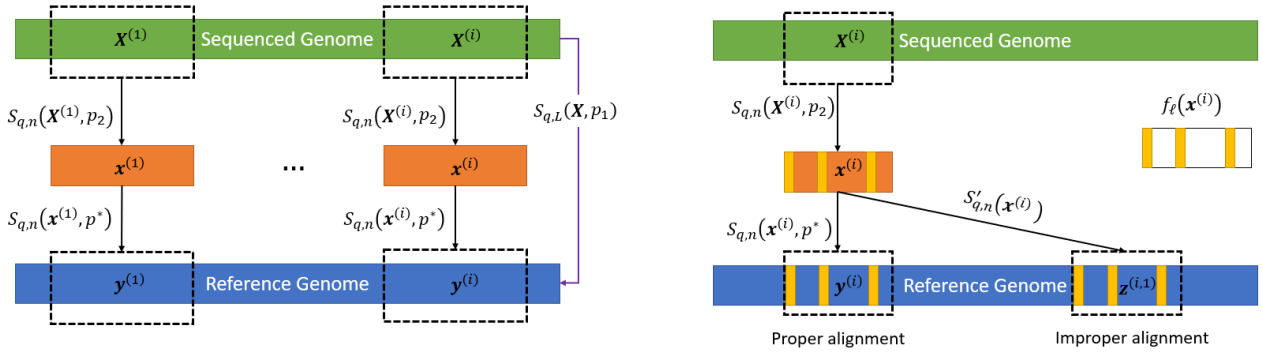

Fig. 1: **Left:** DNA reads generated as fragments of a genome. **Right:** read-identifier-based alignment of a single read.

The channel output string will be denoted by $\mathbf{Y} = \mathsf{S}_{q,L}(\mathbf{X}, p_1)$. Let $\mathbf{X}^{(i)} = X_{k_i}, X_{k_i+1}, \ldots, X_{k_i+n-1}$ be a substring observed by the sequencer, where k_i is a random, unknown *starting index*. The corresponding read $\boldsymbol{x}^{(i)}$ is modeled by $\boldsymbol{x}^{(i)} = \mathsf{S}_{q,n}(\mathbf{X}^{(i)}, p_2)$. Let $\boldsymbol{y}^{(i)} = Y_{k_i}, Y_{k_i+1}, \ldots, Y_{k_i+n-1}$ be the substring of $\mathbf{Y}$ with *proper alignment* to the read $\boldsymbol{x}^{(i)}$. Based on the models above, we get that $\boldsymbol{y}^{(i)} = \mathsf{S}_{q,n}(\boldsymbol{x}^{(i)}, p^*)$, where $p^* \triangleq p_1 \cdot [1 - p_2/(q-1)] + (1-p_1) \cdot p_2$. That is, the differences between reads and their references are modeled as a single substitution channel with parameter p^*. This model can be extended to richer models (e.g. [17]), potentially exploiting the dependence between two overlapping reads, whose references pass through the same instantiation of the channel $\mathsf{S}_{q,L}(\mathbf{X}, p_1)$. The relations between the genome, reference and reads are illustrated in Fig 1.

C. Pre-Decoding Alignment

We wish to encode $\boldsymbol{x}^{(i)}$ such that only the information required to reconstruct it from $\boldsymbol{y}^{(i)}$ is transmitted. Nevertheless, for this reconstruction to work, the decoder first needs to know the starting index k_i of this read. Therefore, the encoder must transmit some additional information enabling the decoder to align the read within the reference, while accounting for the substitution errors. This alignment information is an ℓ-bit *read identifier*, output from a function denoted by $f_\ell(\boldsymbol{x}^{(i)})$. Such identifier and the tradeoffs in setting the value ℓ are discussed in Sections III and IV, respectively.

Since only partial information is provided for alignment, additional *improper alignments*, i.e., erroneous starting indices, are likely to be found. This process provides the decoder with a set $\left\{\boldsymbol{z}^{(i,j)} | j = 1, 2, \ldots\right\}$ of length-n substrings of $\mathbf{Y}$ as candidates for read alignment. Every $\boldsymbol{z}^{(i,j)} \neq \boldsymbol{y}^{(i)}$ can be regarded as having been obtained from $\boldsymbol{x}^{(i)}$ passing through a useless channel with zero mutual information. The alignment process is illustrated in Fig 1 (right). Clearly, only the proper alignment is desired for decoding, thus a method for rejecting false candidates is required. This method, described in Section III, is referred to as *validation*.

III. Code Construction

For simplicity, throughout this section we assume $q = 2$, but the construction can be generalized to any q that is a prime power. Furthermore, cyclic indices will be used, i.e., every index j in $\mathbf{X}, \mathbf{Y}$ will be taken as $([(j-1) \mod L] + 1) \in \{1, \ldots, L\}$, to avoid edge effects.

A. Read Identifier

A simple *bit sampling* approach is found to be very suitable for read identifiers. Let $1 \leq i_1 < \cdots < i_\ell \leq n$ be a predefined set of indices, known to both the encoder and decoder. Now, let $f_\ell(\boldsymbol{x}^{(i)}) = x_{i_1}, \ldots, x_{i_\ell}$ be the read identifier. In this case, an *aligner* at the central node simply correlates this identifier along the reference by evaluating the Hamming distance with respect to each starting index, and produces the set

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

where $d_H(\cdot,\cdot)$ is the Hamming distance, T is a predefined threshold, and $\left\{\boldsymbol{z}^{(i,j)} = \left[Y_{k_i^{(j)}}, \ldots, Y_{k_i^{(j)}+n-1}\right]\right\}$ is the set of possible alignments of $\boldsymbol{x}^{(i)}$ within $\mathbf{Y}$. The remainder of the read, i.e., the indices outside of the identifier, is denoted by $\boldsymbol{x}^{(i)}_{\mathcal{I}}$, where $\mathcal{I} = \{1, \ldots, n\} \setminus \{i_1, \ldots, i_\ell\}, |\mathcal{I}| = n - \ell$.

B. General Code Construction

Our goal is to design a coding scheme for transmitting reads from $\mathbf{X} \in \{0,1\}^L$ such that a decoder with access to $\mathbf{Y} = \mathsf{S}_{2,L}(\mathbf{X}, p^*)$ will be able to perfectly reconstruct them with high probability.

Definition 1. *A $(M, n, \mathcal{R}, p^*, P_\mathsf{s})$-code is a pair $(\mathcal{E}, \mathcal{D})$ of encoder-decoder for a set $\{\boldsymbol{x}^{(i)}\}_{i=1}^M$ of length-n reads such that:*

1) *$\mathcal{E}, \mathcal{D}$ have access only to $\{\boldsymbol{x}^{(i)}\}_{i=1}^M$, $\mathbf{Y}$, respectively,*
2) *the encoded size satisfies*

$$\left|\mathcal{E}\left(\{\boldsymbol{x}^{(i)}\}_{i=1}^M\right)\right| = (nM) \cdot \mathcal{R},$$

3) *the correct decoding probability satisfies*

$$\Pr\left\{\mathcal{D}\left[\mathcal{E}\left(\{\boldsymbol{x}^{(i)}\}_{i=1}^M\right), \mathbf{Y}\right] = \{\boldsymbol{x}^{(i)}\}_{i=1}^M\right\} \geq P_\mathsf{s}.$$

Our general code construction is based on generalized error locating (GEL) codes [15], adapted to use as a source code with alignment-validation capabilities.

Construction 1. 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 denoted by $\bar{\mathsf{H}}_2$, the *validation matrix*. Let H_c be a matrix such that the concatenation of its rows with H_2 forms a square *full-rank* matrix H. This structure is illustrated in Fig. 2.

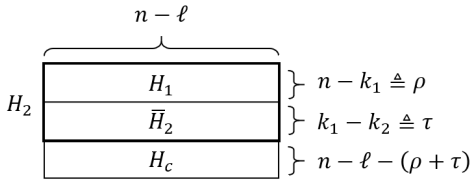

Fig. 2: Structure and sizes of the construction's inner-code parity-check matrices.

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 where $\nu = n - \ell - (\rho+\tau)$. This code will be referred to as the *outer code*.

The encoding and decoding processes are introduced in Algorithms 1 and 2, respectively. We note that the syndrome $\boldsymbol{s}^{(i)}$ with respect to H_2, calculated in Algorithm 1, is of the form $\boldsymbol{s}^{(i)} = [\boldsymbol{s}_1^{(i)}, \boldsymbol{s}_2^{(i)}]$, where $\boldsymbol{s}_1^{(i)}, \boldsymbol{s}_2^{(i)}$ correspond to $\mathsf{H}_1, \bar{\mathsf{H}}_2$, respectively. In Algorithm 2, we denote by $\mathcal{D}_1(\boldsymbol{z}, \boldsymbol{s})$ the result of decoding the word $\boldsymbol{z}$ with respect to H_1 within the coset of syndrome $\boldsymbol{s}$. The same is done 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 $\mathbf{S}$. We denote by $\mathcal{F}_\mathsf{H}(\boldsymbol{u})$ the linear mapping of $\boldsymbol{u}$ to the single codeword of syndrome $\boldsymbol{u}$ in the code defined by H. Finally, we denote by $\bigotimes$ an erasure occurring if either more than one alignment was validated or all of them failed to be validated.

Algorithm 1: Construction 1 Encoding

Input: $\{\boldsymbol{x}^{(i)}\}_{i=1}^M, \mathsf{H}_2, \mathsf{H}_\mathsf{c}, \mathbf{H}_\mathsf{o}$
for $1 \le i \le M$ **do** // Inner Encoding
  Extract $\boldsymbol{w}^{(i)} = f_\ell(\boldsymbol{x}^{(i)})$
  Calculate $\boldsymbol{s}^{(i)} = \mathsf{H}_2 \left[\boldsymbol{x}_\mathcal{I}^{(i)}\right]^T$, $\boldsymbol{a}^{(i)} = \mathsf{H}_\mathsf{c} \left[\boldsymbol{x}_\mathcal{I}^{(i)}\right]^T$
end
Form $\underline{\boldsymbol{a}} = \left[\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(M)}\right] \in [\mathsf{GF}(2^\nu)]^M$
Calculate $\mathbf{S} = \mathbf{H}_\mathsf{o} \underline{\boldsymbol{a}}^T$ // Outer Encoding
Output: $\mathcal{E}\left(\{\boldsymbol{x}^{(i)}\}_{i=1}^M\right) = \left\{\{\boldsymbol{w}^{(i)}\}_{i=1}^M, \{\boldsymbol{s}^{(i)}\}_{i=1}^M, \mathbf{S}\right\}$

Note that the term *validation* in Algorithm 2 and in the sequel includes cases of *misvalidation*, that is, validation of a vector $\boldsymbol{v} \neq \boldsymbol{x}_\mathcal{I}^{(i)}$, whereas a failed validation, i.e. $\hat{\boldsymbol{s}}_2^{(i)} \neq \boldsymbol{s}_2^{(i)}$, forms a *rejection* of the candidate.

Proposition 2. *The rate of Construction 1 is given by*

$$\mathcal{R} = 1 - \frac{k_\mathsf{o}}{M} \cdot \frac{n - \ell - (\rho+\tau)}{n} = 1 - \frac{k_\mathsf{o}}{M} \cdot \frac{k_2 - \ell}{n}.$$

Proposition 3. *Construction 1 yields a $(M, n, \mathcal{R}, p^*, P_\mathsf{s})$-code if and only if at the outer decoder output $\Pr\{\hat{\underline{\boldsymbol{a}}} = \underline{\boldsymbol{a}}\} \ge P_\mathsf{s}$.*

By Proposition 3, the success of the scheme depends on the success in decoding the outer code over a channel induced by the inner-decoding outcomes. In the next section we turn to analyze this induced channel, which requires accounting for both the alignment and the inner-decoder performance.

Algorithm 2: Construction 1 Decoding

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 $1 \le i \le M$ **do**
  Align $\boldsymbol{w}^{(i)}$ over $\mathbf{Y}$, and form $\mathcal{Z}^{(i)}$ (Eq. 1)
  for $1 \le j \le |\mathcal{Z}^{(i)}|$ **do** // Inner Decoding
    Set 'found' $\leftarrow 0$
    Decode $\boldsymbol{v} = \mathcal{D}_1(\boldsymbol{z}_\mathcal{I}^{(i,j)}, \boldsymbol{s}_1^{(i)})$
    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
      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
      end
    end
  end
  if *'found'* $= 0$ **then** // No Candidates
    Set $\boldsymbol{b}^{(i)} = \bigotimes$
  end
end
// Outer Decoding
Decode $\hat{\underline{\boldsymbol{a}}} = \mathcal{D}_\mathsf{o}(\underline{\boldsymbol{b}}, \mathbf{S})$, where $\underline{\boldsymbol{b}} = [\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(M)}]$
for $1 \le i \le M$ **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 $\hat{\boldsymbol{x}}_\mathcal{I}^{(i)}, \boldsymbol{w}^{(i)}$
end
Output: $\{\hat{\boldsymbol{x}}^{(i)}\}_{i=1}^M$

IV. Scheme Analysis

Throughout this section, the sequenced genome is assumed to be a random sequence in which each symbol is drawn i.i.d. from the Bernoulli$(1/2)$ distribution. This assumption is only needed for analyzing the misalignments of the read $\boldsymbol{x}^{(i)}$, and the scheme's handling of the proper alignment does not rely on it. Extension of the analysis to other genome statistical models can be done based on related studies such as [1], [18].

A. Inner-Code Analysis

The inner decoder is invoked in Algorithm 2 on both the proper alignment of $\boldsymbol{x}^{(i)}$ in $\mathbf{Y}$ (if found) and, possibly, on improperly aligned $\boldsymbol{z}^{(i,j)}$ vectors that are not related to $\boldsymbol{x}^{(i)}$. The scheme's performance depends on the inner-decoding outcomes for both types of inputs, which we now analyze.

Definition 4. *Let $\mathcal{Z}^{(i)}$ be the set of possible alignments of some read $\boldsymbol{x}_\mathcal{I}^{(i)}$. The following terms are defined:*

- $P_{\mathsf{a.s}}$ *- the probability of the proper alignment being found, i.e., $\boldsymbol{y}^{(i)} \in \mathcal{Z}^{(i)}$.*
- $\mathsf{K}_\mathsf{f}^{(i)} = \left|\mathcal{Z}^{(i)} \setminus \{\boldsymbol{y}^{(i)}\}\right|$ *- the number of improper alignments.*

Let $\mathsf{F}_\mathsf{b}(n, p, t)$ denote the *CDF* of a binomial random variable with parameters (n, p), evaluated at the value of t.

Lemma 5. *The following expressions are immediate:*

$$P_{\mathsf{a.s}} = \mathsf{F}_\mathsf{b}(\ell, p^*, \mathsf{T}),\ \mathbb{E}[\mathsf{K}_\mathsf{f}^{(i)}] \triangleq \overline{\mathsf{K}_\mathsf{f}} = (L-1) \cdot \mathsf{F}_\mathsf{b}(\ell, 1/2, \mathsf{T}).$$

Definition 6. *For each alignment candidate $\boldsymbol{z}^{(i,j)} \in \mathcal{Z}^{(i)}$ of some read $\boldsymbol{x}_{\mathcal{I}}^{(i)}$, the following probabilities are defined:*

- P_{suc} *- for successful inner decoding, that is,* $\boldsymbol{v} = \boldsymbol{x}_{\mathcal{I}}^{(i)}$.
- P_{miv} *- for misvalidated inner decoding, that is,* $\mathcal{D}_1$ *returns* $\boldsymbol{v} \neq \boldsymbol{x}_{\mathcal{I}}^{(i)}$, *and* $\hat{\boldsymbol{s}}_2^{(i)} = \boldsymbol{s}_2^{(i)}$.
- $P_{\mathsf{rej}} = 1 - P_{\mathsf{suc}} - P_{\mathsf{miv}}$ *- for detected inner-decoding error.*

We further add a superscript $(\mathsf{x}) \in \{(\mathsf{p}), (\mathsf{i.p})\}$ *to the above probabilities, corresponding to whether* $\boldsymbol{z}^{(i,j)}$ *is the proper or improper alignment, respectively. Let* $t_1 \triangleq \lfloor \frac{d_1 - 1}{2} \rfloor$, $n_\ell \triangleq n - \ell$, *and* $V_n(t)$ *denote the volume of a Hamming ball with radius* t *of length-*n *words. Then we have the following.*

Lemma 7. *The following equalities and approximations hold:*

$$P_{\mathsf{suc}}^{(\mathsf{p})} = \mathsf{F}_{\mathsf{b}}(n_\ell, p^*, t_1),\ P_{\mathsf{suc}}^{(\mathsf{i.p})} \simeq 0, \tag{2}$$

$$P_{\mathsf{miv}}^{(\mathsf{i.p})} = \frac{V_{n_\ell}(t_1)}{2^{\rho+\tau}},\ P_{\mathsf{miv}}^{(\mathsf{p})} \lesssim P_{\mathsf{miv}}^{(\mathsf{i.p})}, \tag{3}$$

$$P_{\mathsf{rej}}^{(\mathsf{i.p})} \simeq 1 - P_{\mathsf{miv}}^{(\mathsf{i.p})}. \tag{4}$$

Proof: *In* (2) *the equality follows from* $\mathcal{C}_1$ *having minimum distance* d_1, *and the* $\simeq 0$ *represents the negligible probability that a* $\boldsymbol{z}_{\mathcal{I}}^{(i,j)}$ *would somehow successfully decode to an unrelated* $\boldsymbol{x}_{\mathcal{I}}^{(i)}$. *In the left equality of* (3), *since the improper alignment is not related to the encoder input* $\boldsymbol{x}_{\mathcal{I}}^{(i)}$, $P_{\mathsf{miv}}^{(\mathsf{i.p})}$ *equals the probability that a random length-*n_ℓ *vector has a codeword of* $\mathcal{C}_2$ *at distance at most* t_1. *This can be shown to equal the right-hand side. The* $\lesssim$ *inequality follows from the fact that the probable proximity of* $\boldsymbol{x}_{\mathcal{I}}^{(i)}$ *to the proper alignment in general reduces the probability of misvalidation. For this inequality to hold formally, the inner codes need to satisfy a natural property called properness [19]. Finally,* (4) *follows from the second claim in* (2) *and the definition of* P_{rej}. ∎

B. Outer-Code Analysis

In the outer code, the syndromes with respect to H_{c} are treated as symbols in $\mathsf{GF}(2^\nu)$. As seen in Algorithm 2, these symbols may be erased depending on the outcome of the inner decoding. Furthermore, a misvalidated inner-decoder output $\boldsymbol{v}$ may introduce an erroneous symbol. The outer decoded word $\underline{\boldsymbol{b}}$ can thus be modeled as transmitting the correct codeword $\underline{\boldsymbol{a}}$ through an *outer channel*, which produces erasures and errors. The probabilities of these events are directly induced by the coding scheme parameters $\{\ell, k_1, \tau, \mathsf{T}\}$. We can now examine the erasure and error probabilities of the outer channel, denoted by $P_{\mathsf{ers}}, P_{\mathsf{err}}$, respectively. For the next lemma, we assume that equation (4) holds by equality.

Lemma 8. *Let* $P_{\mathsf{p.a}} \triangleq P_{\mathsf{a.s}} \cdot \left(1 - P_{\mathsf{rej}}^{(\mathsf{p})}\right)$, $P_{\mathsf{p.m}} \triangleq P_{\mathsf{a.s}} \cdot P_{\mathsf{miv}}^{(\mathsf{p})}$, *then the outer channel probabilities satisfy:*

$$\begin{aligned} P_{\mathsf{ers}} &\leq P_{\mathsf{p.a}} \left[\overline{\mathsf{K}}_{\mathsf{f}} P_{\mathsf{miv}}^{(\mathsf{i.p})}(1 + B_1)\right] + (1 - P_{\mathsf{p.a}}) \left[1 - \overline{\mathsf{K}}_{\mathsf{f}} P_{\mathsf{miv}}^{(\mathsf{i.p})}(1 + B_1)\right], \\ P_{\mathsf{err}} &\leq P_{\mathsf{p.m}} \left[1 - \overline{\mathsf{K}}_{\mathsf{f}} P_{\mathsf{miv}}^{(\mathsf{i.p})}(1 + B_1)\right] + (1 - P_{\mathsf{p.a}}) \left[\overline{\mathsf{K}}_{\mathsf{f}} P_{\mathsf{miv}}^{(\mathsf{i.p})}(1 + B_1)\right], \end{aligned} \tag{5}$$

where B_1 *are terms bounded from above by* $\overline{\mathsf{K}}_{\mathsf{f}} P_{\mathsf{miv}}^{(\mathsf{i.p})}$ *when* $\overline{\mathsf{K}}_{\mathsf{f}} P_{\mathsf{miv}}^{(\mathsf{i.p})} < 1$, *and are thus negligible when* $\overline{\mathsf{K}}_{\mathsf{f}} P_{\mathsf{miv}}^{(\mathsf{i.p})} \ll 1$.

Proof: (sketch): For P_{ers} the events of interest are: 1) having 1 proper validation and 1 or more improper validations (first term), and 2) having 0 proper validations and not exactly 1 improper validation (second term). For P_{err} the events of interest are: 1) having 1 proper *mis*-validation and no improper validations (first term), and 2) having 0 proper validations and 1 improper validation (second term). Each probability is expanded with the Taylor series, and B_1 encapsulates only terms of order at least 1 in $\overline{\mathsf{K}}_{\mathsf{f}} P_{\mathsf{miv}}^{(\mathsf{i.p})}$. Since $\mathsf{K}_{\mathsf{f}}^{(i)}$ are themselves binomially distributed, marginalizing the probability over K_{f} gives the raw binomial moments, all of which are order at least 1 in $\overline{\mathsf{K}}_{\mathsf{f}} P_{\mathsf{miv}}^{(\mathsf{i.p})}$ [20]. ∎

We note that relieving the assumption regarding equation (4), the lemma still holds by simply replacing $P_{\mathsf{miv}}^{(\mathsf{i.p})}$ terms with $1 - P_{\mathsf{rej}}^{(\mathsf{i.p})}$.

It is well known that a code with minimum distance d can correct up to m_1 erasures and m_2 errors as long as $m_1 + 2m_2 \leq d - 1$ [21]. That means the redundancy of the outer code should be set by the following.

Proposition 9. *Let* $\mathcal{C}_{\mathsf{o}}$ *be a* maximum distance separable *(MDS) code, and define the random variable* $W = m_1 + 2m_2$. *Then the minimal required redundancy* $M - k_{\mathsf{o}}$ *of* $\mathcal{C}_{\mathsf{o}}$ *equals*

$$\rho_{\mathsf{o}}^* = \min\{w\} \ \textit{ such that } \ P\left(W \leq w\right) \geq P_{\mathsf{s}}\,,$$

where

$$P(W = u) = \sum_{\underline{m} \in S(M,u)} \frac{M!}{m_0! m_1! m_2!} (1 - P_{\mathsf{ers}} - P_{\mathsf{err}})^{m_0} \cdot P_{\mathsf{ers}}^{m_1} \cdot P_{\mathsf{err}}^{m_2},$$

for $\underline{m} = (m_0, m_1, m_2)$ *and*

$$S(M, u) \triangleq \{\underline{m} \mid m_1 + 2m_2 = u\ ,\ m_0 + m_1 + m_2 = M\}\,.$$

V. Parameter Optimization

Given the specified system parameters $M, n, L, p^*, P_{\mathsf{s}}$, to obtain minimal-rate compression by Construction 1 we need to optimize the free parameters of the construction. More concretely, the optimal parameters are defined in the following.

Definition 10. *The* optimal parameters *of a* $(M, n, \mathcal{R}, p^*, P_{\mathsf{s}})$*-code from Construction 1 are*

$$\{k_1, \tau, \ell, \mathsf{T}\}^\star = \underset{\{k_1, \tau, \ell, \mathsf{T}\}}{\arg\max}\, \mathcal{L}(k_1, \tau, \ell, \mathsf{T}), \tag{6}$$

where

$$\mathcal{L}(k_1, \tau, \ell, \mathsf{T}) \triangleq (k_1 - \tau - \ell) \cdot (M - \rho_{\mathsf{o}}^*). \tag{7}$$

The optimality of $\{k_1, \tau, \ell, \mathsf{T}\}^\star$ in Definition 10 follows immediately from Propositions 2 and 9. The challenge in finding the optimal values lies in that maximizing (7) needs to account for the complex dependence of ρ_{o}^* on the other parameters affecting the rate.

A. Choosing the Alignment Parameters

Finding the four parameters of (6) simultaneously is difficult, so we first deal with the alignment parameters ℓ,T separately. We observe that these parameters have importance beyond the above optimization: they control the expected number of improper alignments, and thus the decoding complexity (the expected number of inner-decoder invocations).

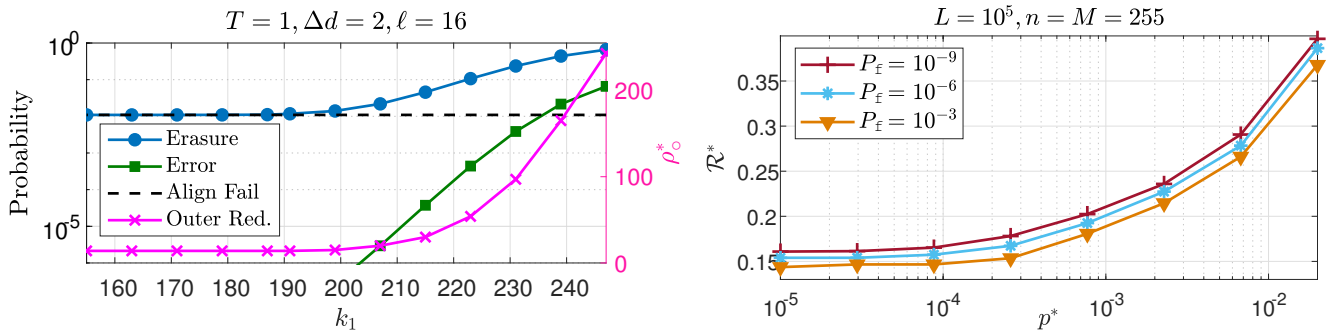

Fig. 3: Example of Construction 1 using BCH inner codes for $L = 10^5, n = M = 255$. **Left:** outer channel erasure and error probabilities and required outer redundancy symbols. **Right:** optimal rate $\mathcal{R}^*$ as a function of p^* for three failure probabilities.

Therefore, we limit ourselves to values that maintain a limit on this expected number, as follows.

Proposition 11. *Given an upper bound* $\mathsf{K_m}$ *for the expected number of improper alignments* $\overline{\mathsf{K}}_\mathsf{f}$*, a valid region for threshold* T *values, for every read identifier length* ℓ*, is obtained by*

$$\mathsf{T_V}(\mathsf{K_m}, \ell) = \left\{\mathsf{T} : \; V_\ell(\mathsf{T}) \cdot 2^{-\ell} \leq \mathsf{K_m}/L\right\}.$$

The valid region for ℓ *is then* $\ell_\mathsf{V}(\mathsf{K_m}) = \{\ell : \mathsf{T_V}(\mathsf{K_m}, \ell) \neq \emptyset\}$.

The pairs ℓ,T allowed by Proposition 11 are independent of all other (system+construction) parameters except L, and so can be calculated once and reused for every parameter setup.

B. Approximating the Outer Redundancy

For efficient evaluation of parameter sets during optimization, it is helpful to simplify the dependence of ρ_o^* on P_ers,P_err relative to the exact trinomial CDF in Proposition 9. For that, we approximate the trinomial distribution of W with parameters $M, P_\mathsf{ers}, P_\mathsf{err}$ by a Normal distribution with the same mean and same variance, as follows.

Lemma 12. *Let* $P_\mathsf{red} \triangleq P_\mathsf{ers} + 2P_\mathsf{err}$*. The outer redundancy can be approximated by*

$$\rho_\mathsf{o}^* \approx \mu_W + Q^{-1}(P_\mathsf{f}) \cdot \sigma_W + \beta(P_\mathsf{f}), \tag{8}$$

where $\mu_W = MP_\mathsf{red}$, $\sigma_W^2 = MP_\mathsf{red}(1 - P_\mathsf{red}) + 2MP_\mathsf{err}$*, and* $Q(\cdot)$ *is the Normal distribution's tail function,* $P_\mathsf{f} = 1 - P_\mathsf{s}$ *is the allowed failure probability, and* $\beta(\cdot)$ *is some empirical correction function of* P_f*.*

The advantage of (8) is that ρ_o^* can be approximated in closed form, using a single fitting parameter β for each P_f, and can be used for any pair P_ers,P_err induced by inner-code parameters.

C. Optimization Procedure

Finding a solution to the problem in (6) can now be done in the following stages: (i.) Given $\mathsf{K_m}$, find the valid regions $\ell_\mathsf{V}(\mathsf{K_m})$,$\mathsf{T_V}(\mathsf{K_m}, \ell)$, (ii.) set k_1 and iterate over valid pairs (ℓ, T), and numerically find the optimal $\tau^\star$ by evaluating equations (5), (7) and (8) (neglecting B_1 terms), (iii.) find the optimal set $\{\tau, \ell, \mathsf{T}\}^\star$ for k_1, by comparing the results of each pair (ℓ, T), and (iv.) find the optimal set $\{k_1, \tau, \ell, \mathsf{T}\}^\star$ by comparing the results of each k_1. Since the set of available dimensions k_1 of an error-correcting code family (e.g. BCH) is relatively small, the search space in (iv) is manageable. Similarly, the search space of $\tau^\star$ in (ii) is relatively small, since the misvalidation probability decays exponentially with the number of validation bits.

VI. Numerical Results

We demonstrate the optimization procedure by showing numerical results for Construction 1 employing binary BCH codes as $\mathcal{C}_1, \mathcal{C}_2$, and a Reed-Solomon code as $\mathcal{C}_\mathsf{o}$. In this setting, the parameter τ is obtained by choosing the difference in minimum distance between $\mathcal{C}_1$ and $\mathcal{C}_2$, denoted $\Delta d \triangleq d_2 - d_1$. In Fig. 3 (left), we plot the induced outer channel probabilities (left axis) and corresponding outer redundancy (right axis) as a function of k_1, for $L = 10^5, n = M = 255, p^* = 0.01, P_\mathsf{s} = 1 - 10^{-6}$. The tradeoff of redundancy allocation between codes is clear by examining the (non-linear) increase in required outer redundancy with the weakening of the inner code (increase in k_1). The plot reveals a floor value of 10^{-2} for the erasure probability, attributed to failed proper alignment, which cannot be solved by strengthening the inner code. A solution of the rate optimization problem for different values of substitution rate p^* is shown for $P_\mathsf{f} = \{10^{-9}, 10^{-6}, 10^{-3}\}$ in Fig. 3 (right). Finally, we compare in Fig. 4 the rates of the proposed scheme with a powerful fixed-rate benchmark that assumes the encoder knows both $\mathbf{Y}$ and the correct alignment of every $\boldsymbol{x}^{(i)}$ in $\mathbf{Y}$ (and can thus communicate the alignment index plus a fixed-rate encoding of the difference $\boldsymbol{y}^{(i)} - \boldsymbol{x}^{(i)}$). It is shown that for $P_\mathsf{f} = 10^{-9}$, the proposed scheme outperforms the benchmark for every p^* starting at 10^{-4}, while saving the space and computation effort at the encoder required to store $\mathbf{Y}$ and align the reads to it.

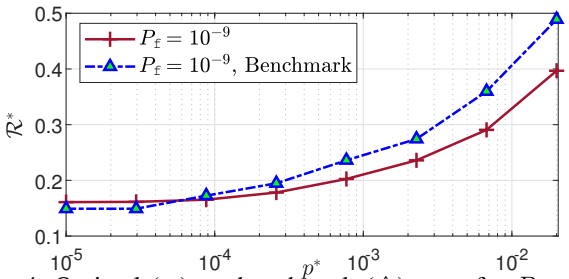

Fig. 4: Optimal (+) vs. benchmark (△) rates for $P_\mathsf{f} = 10^{-9}$.

VII. Acknowledgement

This work was supported in part by the US-Israel Binational Science Foundation and by the Israel Science Foundation.

REFERENCES

[1] A. S. Motahari, G. Bresler, and D. N. C. Tse, "Information Theory of DNA Shotgun Sequencing," *IEEE Trans. on Information Theory, Vol. 59, No. 10*, 2013.

[2] Z. Zhu, Y. Zhang, Z. Ji, S. He, and X. Yang, "High-Throughput DNA Sequence Data Compression," *Briefings In Bioinformatics, Vol. 16, No. 1. 1-15*, 2013.

[3] N. S. Bakr and A. A. Sharawi, "DNA Lossless Compression Algoirthms: Review," *American Journal of Bioinformatics Research, 3(3)*, 2013.

[4] M. Hosseini, D. Pratas, and A. J. Pinho, "A Survey on Data Compression Methods for Biological Sequences," *Information 7, 56*, 2016.

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

[6] 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.

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

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

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

[10] A. Aaron and B. Girod, "Compression with Side Information Using Turbo Codes," *IEEE Proceedings DCC 2002. Data Compression Conference*, 2002.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] M. Bossert, "Channel Coding for Telecommunications," *Wiley*, 1999.

[16] A. F. Wright, "Genetic Variation: Polymorphisms and Mutations," *Encyclopedia of Life Sciences, John Wiley and Sons*, 2005.

[17] C. Lottaz, C. Iseli, C. Jongeneel, and P. Bucher, "Modeling Sequencing Errors by Combining Hidden Markov Models," *Bioinformatics, Vol. 19, Suppl. 2*, 2003.

[18] C. Kusters and T. Ignatenko, "DNA Sequence Modeling Based On Context Trees," *Proc. of the 36th WIC Symposium on Information Theory*, 2015.

[19] S. K. Leung-Yan-Cheong, E. R. Barnes, and D. U. Friedman, "On Some Properties of the Undetected Error Probability of Linear Codes," *IEEE Trans. on Information Theory, Vol. IT-25, No. 1*, 1979.

[20] A. Knoblauch, "Closed-Form Expression for the Moments of the Binomial Probability Distribution," *SIAM Journal on Applied Mathematics, Vol. 69, No. 1*, 2008.

[21] R. M. Roth, "Introduction To Coding Theory," *Cambridge University Press*, 2006.