Single Indel/Edit Correcting Codes: Linear-Time Encoders and Order-Optimality

Kui Cai, Yeow Meng Chee, Ryan Gabrys, Han Mao Kiah, and Tuan Thanh Nguyen

***Abstract*—An indel refers to a single insertion or deletion, while an edit refers to a single insertion, deletion or substitution. In this work, we investigate quaternary codes that correct a single indel or single edit and provide linear-time algorithms that encode binary messages into these codes of length n. Particularly, we provide two linear-time encoders: one corrects a single edit with $\lceil \log n \rceil + O(\log \log n)$ redundancy bits, while the other corrects a single indel with $\lceil \log n \rceil + 2$ redundant bits. These two encoders are *order-optimal*. The former encoder is the first known order-optimal encoder that corrects a single edit, while the latter encoder (that corrects a single indel) reduces the redundancy of the best known encoder of Tenengolts (1984) by at least four bits.**

I. Introduction

Correcting deletions, insertions, and substitutions plays an essential role in improving the reliability of DNA-based storage systems or file synchronization. In a DNA-based storage system, the input user data is translated into a large number of DNA strands, which are synthesized and stored in a DNA pool. To retrieve the original data, the stored DNA strands are sequenced and translated inversely back to the binary data. It has been found that substitutions, deletions, and insertions are most common errors occurring at the stages of synthesis and sequencing [2]–[4]. Given that current synthesis technologies produce strands of lengths 100 to 200 and given the low raw error rates reported by experiments (see, for example, [3], [4]), we expect most DNA strands to be corrupted by at most one edit error. In this work, we focus on codes that combat either a *single indel* or a *single edit* and provide efficient methods of encoding binary messages into these codes.

Now, to correct a single indel, we have the celebrated class of Varshamov-Tenengolts (VT) codes. While Varshamov and Tenengolts introduced the binary version to correct asymmetric errors [8], Levenshtein later modified the VT construction to correct a single edit [5]. In both constructions, the number of redundant bits is $\log n + O(1)$, where n is the length of a codeword. Unless otherwise stated, all logarithms are taken base two.

A nonbinary version of the VT codes was proposed by Tenengolts [6], who also provided a linear-time method to correct a single indel. In the same paper, Tenengolts also presented an efficient encoder that corrects a single indel. For the quaternary alphabet, this encoder requires at least $\log n + 7$ bits for words of length $n \geqslant 20$. To the best of our knowledge, there is no known efficient construction for q-ary codes that can correct a single edit.

In summary, our broad objective is to provide practical quaternary codes that correct either a single indel or a single edit. Our coding techniques can be applied to construct q-ary codes for $q = \Theta(\log n)$. Due to space constraints, we only summarise the results and describe the main idea of our constructions. More details can be found in [1].

II. Preliminary

In this work, we use the following relation Φ between the decimal alphabet $\Sigma = \{0, 1, 2, 3\}$ and the nucleotides $\mathcal{D} = \{\mathtt{A}, \mathtt{T}, \mathtt{C}, \mathtt{G}\}$, $\Phi : 0 \to \mathtt{A}, 1 \to \mathtt{T}, 2 \to \mathtt{C}$, and $3 \to \mathtt{G}$.

Let $\boldsymbol{x} \in \Sigma^n$. We are interested in the following *error balls*:

$$\mathcal{B}^{\rm indel}(\boldsymbol{x}) \triangleq \{\boldsymbol{x}\} \cup \{\boldsymbol{y} : \boldsymbol{y} \text{ is obtained from } \boldsymbol{x} \text{ via a single indel}\},$$
$$\mathcal{B}^{\rm edit}(\boldsymbol{x}) \triangleq \{\boldsymbol{x}\} \cup \{\boldsymbol{y} : \boldsymbol{y} \text{ is obtained from } \boldsymbol{x} \text{ via a single edit}\}.$$

Let $\mathcal{C} \subseteq \Sigma^n$. We say that $\mathcal{C}$ *corrects a single indel* if $\mathcal{B}^{\rm indel}(\boldsymbol{x}) \cap \mathcal{B}^{\rm indel}(\boldsymbol{y}) = \varnothing$ for all distinct $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$. Similarly, $\mathcal{C}$ *corrects a single edit* if $\mathcal{B}^{\rm edit}(\boldsymbol{x}) \cap \mathcal{B}^{\rm edit}(\boldsymbol{y}) = \varnothing$ for all distinct $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$. Our design objective is to minimize the number of redundant bits. Particularly, the redundancy is $K \log n + o(\log n)$, where K is a constant to be minimized. When $K = 1$, we say that the code is *order-optimal*.

We now review literature works. The *binary VT syndrome* of a binary sequence $\boldsymbol{x} \in \{0,1\}^n$ is defined to be $\mathrm{Syn}(\boldsymbol{x}) = \sum_{i=1}^{n} i x_i$. For binary codes, Levenshtein [5] constructed the following codes:

$$\mathrm{VT}_a(n) = \{\boldsymbol{x} \in \{0,1\}^n : \mathrm{Syn}(\boldsymbol{x}) = a \pmod{n+1}\}.$$
$$\mathrm{L}_a(n) = \{\boldsymbol{x} \in \{0,1\}^n : \mathrm{Syn}(\boldsymbol{x}) = a \pmod{2n}\},$$

and showed that $\mathrm{VT}_a(n)$ can correct a single indel and $\mathrm{L}_a(n)$ can correct a single edit. The constructed codes are both order-optimal.

In 1984, Tenengolts [6] generalized the binary VT codes to nonbinary ones. Tenengolts defined the *signature* of a q-ary vector $\boldsymbol{x}$ of length n to be the binary vector $\alpha(\boldsymbol{x})$ of length $n-1$, where $\alpha(x)_i = 1$ if $x_{i+1} \geq x_i$, and 0 otherwise, for $i \in [n-1]$. For $a \in \mathbb{Z}_n$ and $b \in \mathbb{Z}_q$, set

$$\mathrm{T}_{a,b}(n;q) \triangleq \Big\{\boldsymbol{x} : \ \alpha(\boldsymbol{x}) \in \mathrm{VT}_a(n-1) \text{ and } \sum_{i=1}^{n} x_i = b \pmod{q}\Big\}.$$

Kui Cai and Tuan Thanh Nguyen are with the Singapore University of Technology and Design, Singapore 487372 (email: {cai_kui, tuanthanh_nguyen}@sutd.edu.sg). The work of Kui Cai and Tuan Thanh Nguyen is supported by Singapore Ministry of Education Academic Research Fund Tier 2 MOE2019-T2-2-123.

Yeow Meng Chee is with the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 119077 (email: pvocym@nus.edu.sg).

Ryan Gabrys is with the Spawar Systems Center, San Diego, CA 92152 USA (email: ryan.gabrys@navy.mil).

Han Mao Kiah is with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 639798 (email: hmkiah@ntu.edu.sg).

The results of this work have been accepted to appear in the Special Issue of IEEE Transactions on Information Theory Dedicated to the Memory of Vladimir I. Levenshtein. The preprint version is available on arXiv [1].

Tenengolts showed that $T_{a,b}(n;q)$ corrects a single indel and these codes are order-optimal [6].

Surprisingly, to correct a single edit for the q-ary case, it is not straightforward as in the binary case. One possible strategy is to adapt Levenshtein's method by changing the modulo value in Tenengolts' nonbinary single-deletion correcting codes[1]. Unfortunately, it is not possible to adopt this strategy for Tenengolts' codes. The reason is that it is possible for two distinct words that differ in a single position to share the same signature. For example, consider the ternary words $\boldsymbol{x} = (2,2,2,1)$ and $\boldsymbol{x}' = (2,2,2,0)$. Their signatures are both $\alpha(\boldsymbol{x}) = \alpha(\boldsymbol{x}') = (1,1,0)$ and thus, any syndromes computed based on their signatures result in the same value. Hence, a novel strategy is required and we provide one.

III. Codes Construction

A. Correcting a Single Indel

Instead of Tenengolts' quaternary codes, we investigate a class of binary codes by Levenshtein that corrects a burst of errors. With suitable modifications, we present a linear-time quaternary encoder that corrects a single indel with $\lceil \log n \rceil + 2$ bits of redundancy.

We consider the binary representation of symbols in Σ_4 as follows: $0 \leftrightarrow 00, \quad 1 \leftrightarrow 01, \quad 2 \leftrightarrow 10, \quad 3 \leftrightarrow 11$. Therefore, given a quaternary sequence $\boldsymbol{\sigma} \in \Sigma_4^n$, we have a corresponding binary sequence $\boldsymbol{x} \in \{0,1\}^{2n}$ and we write $\boldsymbol{x} = \Psi(\boldsymbol{\sigma})$. We observed that when an indel occurs in $\boldsymbol{\sigma} \in \mathcal{D}^n$, the binary sequence $\Psi(\boldsymbol{\sigma})$ has a burst of indels of length two. In other words, we are interested in binary codes that correct a single burst of indels of length two. To do so, we have the following construction by Levenshtein [5].

For $\boldsymbol{x} \in \{0,1\}^n$, we write $\boldsymbol{x}$ as the concatenation of s substrings $\boldsymbol{x} = \boldsymbol{u}_0\boldsymbol{u}_1 \ldots \boldsymbol{u}_{s-1}$, where each substring $\boldsymbol{u}_i$ contains identical bits, while substrings $\boldsymbol{u}_i$ and $\boldsymbol{u}_{i+1}$ contain different bits. Each substring $\boldsymbol{u}_i$ is also known as a *run* in $\boldsymbol{x}$. Let r_i be the length of the run $\boldsymbol{u}_i$. The *run-syndrome* of the binary word $\boldsymbol{x}$, denoted by $\mathrm{Rsyn}(\boldsymbol{x})$, is defined as $\mathrm{Rsyn}(\boldsymbol{x}) = \sum_{i=1}^{s-1} i r_i$.

Theorem 1 (Levenshtein [7]). *For $a \in \mathbb{Z}_{2n}$, set*

$$\mathrm{L}_a^{\mathrm{burst}}(n) \triangleq \{\boldsymbol{x} \in \{0,1\}^n : \mathrm{Rsyn}(0\boldsymbol{x}) = a \pmod{2n}\}.$$

Then code $\mathrm{L}_a^{\mathrm{burst}}(n)$ can correct a burst of indel of length two.

Our contribution is to design efficient encoders of $\mathrm{L}_a^{\mathrm{burst}}(n)$ for arbitrary a, n, and consequently design an efficient method to translate binary sequences into quaternary codes that can correct a single indel. We then proceed to extend and generalize this construction so as to design efficient encoder for codes capable of correcting a burst of indel errors with $\log n + O(\log\log n)$ bits of redundancy. Details can be found in [1].

B. Correcting a Single Edit

A key ingredient of our code construction is the set of all *k-sum-balanced* words.

Definition 1. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \Sigma_4^n$. A window $\boldsymbol{W}$ of length k of $\boldsymbol{x}$, i.e. $\boldsymbol{W} = (x_{i+1}, \ldots, x_{i+k})$ is called *sum-balanced* if $k < \sum_{x_j \in \boldsymbol{W}} x_j < 2k$. A word $\boldsymbol{x}$ is *k-sum-balanced* if every window $\boldsymbol{W}$ of the word $\boldsymbol{x}$ is sum-balanced whenever the window length is at least k.

Set $\mathrm{Bal}_k(n) \triangleq \{\boldsymbol{x} \in \Sigma_4^n : \boldsymbol{x} \text{ is } k\text{-sum-balanced}\}$. We have the following properties of $\mathrm{Bal}_k(n)$.

Lemma 1. *For sufficient large n, if $k = 36 \log n$, then $|\mathrm{Bal}_k(n)| \geq 4^{n-1}$.*

The lemma states that whenever $k = \Omega(\log n)$, the set $\mathrm{Bal}_k(n)$ incurs at most one symbol of redundancy.

Construction 1. *Given $k < n$, set $P = 5k$. For $a \in \mathbb{Z}_{4n+1}$, $b \in \mathbb{Z}_P$, $c \in \mathbb{Z}_2$ and $d \in \mathbb{Z}_7$, set $\mathcal{C}^B(n; a, b, c, d)$ as follows.*

$$\begin{aligned} &\mathcal{C}^B(n; k, a, b, c, d) \\ &\quad = \Big\{\boldsymbol{x} \in \mathrm{Bal}_k(n) : \mathrm{Syn}(\boldsymbol{x}) = a \pmod{4n+1}, \\ &\quad\quad \alpha(\boldsymbol{x}) \in \mathrm{SVT}_{b,c,P}(n-1), \text{ and } \sum_{i=1}^{n} x_i = d \pmod 7\Big\}. \end{aligned}$$

Theorem 2. *The code $\mathcal{C}^B(n; a, b, c, d)$ corrects a single edit in linear-time. There exist a, b, c, d such that the size of $\mathcal{C}^B(n; a, b, c, d)$ is at least*

$$|\mathcal{C}(n; a, b, c, d)| \geq \frac{|\mathrm{Bal}_k(n)|}{35(4n+1)k}.$$

When $k = 36 \log_4 n$, we have that the redundancy is at most $\log_4 n + O(\log\log n)$ bits.

We prove the correctness of Construction 1 and Theorem 2 by providing an efficient decoder that can correct a single edit error in linear time (refer to Lemma 28, Lemma 30 in [1]). A high-level description of the efficient encoder that encodes binary messages into a quaternary codebook that corrects a single edit is also presented in [1].

References

[1] K. Cai, Y. M. Chee, R. Gabrys, H. M. Kiah, and T. T. Nguyen, "Optimal Codes Correcting a Single Indel / Edit for DNA-Based Data Storage", to appear, accepted in *IEEE Transactions on Information Theory, Special Issue Dedicated to the Memory of Vladimir I. Levenshtein*, Nov. 2020. Available on *arXiv*, arXiv:1910.06501.

[2] S. Yazdi, H. M. Kiah, E. R. Garcia, J. Ma, H. Zhao, and O. Milenkovic, "DNA-based storage: Trends and methods", *IEEE Trans. Molecular, Biological, Multi-Scale Commun.*, vol. 1, no. 3, pp. 230–248, 2015.

[3] L. Organick, S. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Racz, G. Kamath, P. Gopalan, B. Nguyen, C. Takahashi, S. Newman, H.-Y. Parker, C. Rashtchian, K. Stewart, G. Gupta, R. Carlson, J. Mulligan, D. Carmean, G. Seelig, L. Ceze, and K. Strauss, "Random access in large-scale DNA data storage", *Nature Biotechnology*, vol. 36, no. 3, 242–248, 2018.

[4] R. Heckel, G. Mikutis, and R. N. Grass, "A Characterization of the DNA Data Storage Channel", *Scientific Reports*, Jul. 2019.

[5] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals", *Doklady Akademii Nauk SSSR*, vol. 163, no. 4, pp. 845-848, 1965.

[6] G. Tenengolts, "Nonbinary codes, correcting single deletion or insertion", *IEEE Transactions on Information Theory*, vol. 30, no. 5, pp. 766-769, 1984.

[7] V. I. Levenshtein, "Asymptotically optimum binary code with correction for losses of one or two adjacent bits", *Systems Theory Research (translated from Problemy Kibernetiki)*, vol. 19, pp. 293-298, 1967.

[8] R. R. Varshamov and G. M. Tenengolts, "Codes which correct single asymmetric errors", *Automatica i Telemekhanica*, vol. 26, no. 2, pp. 288-292, 1965.

[1] In the binary case, Levenshtein increased the modulo value from $(n+1)$ to modulo $2n$ to correct a single substitution.