Single-Deletion Single-Substitution Correcting Codes

Ilia Smagloy[1*], **Lorenz Welter**[1†], **Antonia Wachter-Zeh**[†], and **Eitan Yaakobi**[*],
[*]Technion — Israel Institute of Technology, Israel, {ilia.smagloy, yaakobi}@cs.technion.ac.il
[†]Technical University of Munich, Germany, {lorenz.welter, antonia.wachter-zeh}@tum.de

***Abstract*—Correcting insertions/deletions as well as substitution errors simultaneously plays an important role in DNA-based storage systems as well as in classical communications. This paper deals with the fundamental task of constructing codes that can correct a single insertion or deletion along with a single substitution. A non-asymptotic upper bound on the size of single-deletion single-substitution correcting codes is derived, showing that the redundancy of such a code of length n has to be at least $2\log n$. The bound is presented both for binary and non-binary codes while an extension to single deletion and multiple substitutions is presented for binary codes. An explicit construction of single-deletion single-substitution correcting codes with at most $6\log n + 8$ redundancy bits is derived. Note that the best known construction for this problem has to use 3-deletion correcting codes whose best known redundancy is roughly $24\log n$.**

I. Introduction

Codes correcting insertions/deletions recently attract a lot of attention due to their relevance in DNA-based data storage systems, cf. [1]. In classical communications, insertions/deletions happen during the synchronization of files and symbols of data streams [2] or due to over-sampling and under-sampling at the receiver side [3]. The algebraic concepts correcting insertions and deletions date back to the 1960s when Varshamov and Tenengolts designed a class of binary codes, nowadays called *VT codes*. These codes were originally designed to correct asymmetric errors in the Z-channel [4], [5] and later proven to be able to correct a single insertion or a single deletion [6]. VT codes are asymptotically optimal length-n single-insertion/deletion correcting codes of redundancy $\log(n+1)$. As a generalization of VT codes, Tenengolts presented q-ary single-insertion/deletion correcting codes in [4]. Levenshtein has also proven that for correcting t insertions/deletions, the redundancy is asymptotically at least $t\log n$. In [7], Brakensiek *et al.* presented binary multiple-insertion/deletion correcting codes with small asymptotic redundancy. For an explicit small number of deletions, their construction however needs redundancy $c\log n$ where c is a large constant. The recent parallel works by Gabrys *et al.* [8] and Sima *et al.* [9] have presented constructions to correct two deletions with redundancy $8\log n + O(\log\log n)$ [8] and $7\log n + o(\log n)$ [9], respectively. Sima and Bruck [10] generalized their construction to correct any t insertions/deletions with redundancy $8t\log n + o(\log n)$.

However, in DNA data storage as well as in file/symbol synchronization, not only insertions/deletions occur, but also classical substitution errors. Clearly, a substitution error can be seen as a deletion followed by an insertion. Therefore, in order to correct for example a single deletion and a single substitution, the best known construction uses codes correcting *three* deletions. The leading construction for three-deletion correcting codes is the one by Sima and Bruck [10] which has redundancy roughly $24\log n$.

In this paper, we initiate the study of codes correcting substitutions and insertions/deletions. One of our main results is a construction of a single-deletion single-substitution correcting code with redundancy at most $6\log n + 8$ which significantly improves upon using 3-deletion correcting codes. We also derive a non-asymptotic upper bound on the cardinality of q-ary single-deletion single-substitution codes which shows that at least redundancy $2\log n$ is necessary (in contrast to a 3-deletion correcting code that requires redundancy at least $3\log n$). In the binary case this bound is also generalized to single-deletion multiple-substitution correcting codes.

II. Definitions and Preliminaries

This section formally defines the codes and notations that will be used throughout the paper. For two integers $i, j \in \mathbb{N}$ such that $i \leqslant j$ the set $\{i, i+1, \ldots, j\}$ is denoted by $[i, j]$ and in short $[j]$ if $i = 0$. The alphabet of size q is denoted by $\Sigma_q = \{0, 1, \ldots, q-1\}$. A *t-indel* is any combination of t_D deletions and t_I insertions such that $t_D + t_I = t$. Moreover, for two positive integers, $t \leqslant n$ and $s \leqslant n - t$, $B^{DS}_{t,s}(\boldsymbol{x})$ is the set of all words received from $\boldsymbol{x} \in \Sigma_q^n$ after t deletions and at most s substitutions. Note that the order in which the errors occur does not matter and thus we will mostly assume that first the deletions occurred. Finally, $r(\boldsymbol{x})$ denotes the number of runs in $\boldsymbol{x} \in \Sigma_q^n$.

A code $\mathcal{C} \subseteq \Sigma_q^n$ is called a *t-deletion s-substitution correcting code* if it can correct any combination of at most t deletions and s substitutions. That is, for all $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{C}$ it holds that $B^{DS}_{t,s}(\boldsymbol{c}_1) \cap B^{DS}_{t,s}(\boldsymbol{c}_2) = \emptyset$. We define similarly *t-indel s-substitution correcting code* to be a code that corrects any combination of at most t indels and s substitutions.

The goal of this paper is to study codes correcting indels and substitutions. Similarly to the equivalence between insertion and deletion correcting codes, the following lemma holds.

[1]The first two authors contributed equally to this work. The work of L. Welter and A. Wachter-Zeh has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 801434). The work of I. Smagloy and E. Yaakobi was partially supported by the United States-Israel BSF grant 2018048.

Lemma 1. *A code $\mathcal{C}$ is a t-indel s-substitution correcting code if and only if it is a t-deletion s-substitution-correcting code.*

Therefore, the main focus of the paper is on t-deletion s-substitution correcting codes and specifically for $t = 1$. The size of the largest q-ary length-n single-deletion s-substitution correcting code is denoted by $DS_{s,q}(n)$.

III. Bounds

The method used to compute a non-asymptotic upper bound for the cardinality of any single-deletion s-substitution code is described in [11] and [12]. For clarity of the results, the principal concepts of this method are briefly reviewed. The main idea is to construct a hypergraph $\mathcal{H}_s(X, \mathcal{E}_s)$ out of the channel graph with vertices $X = \Sigma_q^{n-1} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ and hyperedges $\mathcal{E}_s = \{E_1, \ldots, E_\ell\} = \{B_{1,s}^{DS}(\boldsymbol{x}) : \boldsymbol{x} \in \Sigma_q^n\}$. The objective is to find the smallest size of a transversal $T \subseteq X$ in $\mathcal{H}_s$, i.e. T intersects all hyperedges in $\mathcal{H}_s$. Let $\mathbf{I}$ be the $m \times \ell$ incidence matrix of $\mathcal{H}$ where $\mathbf{I}(i,j) = 1$ if $\boldsymbol{x}_i \in E_j$. A transversal $\boldsymbol{w} \in \Sigma_2^m$ satisfies that $\mathbf{I}^T \cdot \boldsymbol{w} \geqslant 1$. If $\boldsymbol{w} \in (\mathbb{R}^+)^m$, then it is called a fractional transversal. Thus, the objective is to find some $w_{\boldsymbol{y}} \geqslant 0$, which needs to fulfill the condition $\sum_{\boldsymbol{y} \in B_{1,s}^{DS}(\boldsymbol{x})} w_{\boldsymbol{y}} \geqslant 1$ for all $\boldsymbol{x} \in \Sigma_q^n$. Consequently, the following expression is an upper bound of the cardinality of a code,

$$|\mathcal{C}| \leqslant \sum_{\boldsymbol{y} \in \Sigma_q^{n-1}} w_{\boldsymbol{y}}.$$

A. Upper Bound on Single-Deletion Single-Substitution Codes

Before determining valid fractional transversals, an important property for any $\boldsymbol{y} \in B_{1,s}^{DS}(\boldsymbol{x})$ is studied in the following.

Claim 1. *For all $\boldsymbol{x} \in \Sigma_q^n$ and $\boldsymbol{y} \in B_{1,s}^{DS}(\boldsymbol{x})$, it holds that*

$$r(\boldsymbol{x}) - (2 + 2s) \leqslant r(\boldsymbol{y}) \leqslant r(\boldsymbol{x}) + 2s.$$

Thus, the monotonicity argument $|B_{1,s}^{DS}(\boldsymbol{y})| \leqslant |B_{1,s}^{DS}(\boldsymbol{x})|$ as in the single deletion case of [11] does not necessarily hold and choosing $w_{\boldsymbol{y}} = \frac{1}{|B_{1,s}^{DS}(\boldsymbol{y})|}$ does not suffice as a feasible solution.

The following lemma extends the result from [13] and provides the size of $B_{1,1}^{DS}(\boldsymbol{x})$ for the q-ary alphabet.

Lemma 2. *For any word $\boldsymbol{x} \in \Sigma_q^n$,*

$$|B_{1,1}^{DS}(\boldsymbol{x})| = \begin{cases} (n-1)(q-1)+1 & r(\boldsymbol{x}) = 1, \\ r(\boldsymbol{x})\left[(n-3)(q-1)+(q-2)\right]+(q+2) & \\ & r(\boldsymbol{x}) \geqslant 2. \end{cases}$$

Now, from the results of Lemma 2 and Claim 1, a valid expression of a fractional transversal $w_{\boldsymbol{y}}$ can be derived.

Lemma 3. *The following choice of $w_{\boldsymbol{y}}$ for $\boldsymbol{y} \in \Sigma_q^{n-1}, n \geqslant 3$, is a fractional transversal for $\mathcal{H}_1$*

$$w_{\boldsymbol{y}}^1 = \begin{cases} \frac{1}{(n-1)(q-1)+1} & r(\boldsymbol{y}) \leqslant 3 \\ \frac{1}{(r(\boldsymbol{y})-2)[(n-3)(q-1)+(q-2)]+(q+2)} & r(\boldsymbol{y}) > 3. \end{cases}$$

The following claim will be used in computing the upper bound of the cardinality of the code.

Claim 2. *For integers $q \geqslant 2$, $n \geqslant 5$, and $n \geqslant q$ it holds that*

$$\sum_{k=1}^{n} \binom{n}{k} (q-1)^k \frac{1}{k} \leqslant \frac{q^{n+1}}{(q-1)(n-2)}.$$

Note that the claim is combined from similar statements in [14, Lemma 13, 14]. Putting everything together the following upper bound on $DS_{1,q}(n)$ can be presented.

Theorem 4. *For $q \leqslant n, n \geqslant 6$ the following is an upper bound on $DS_{1,q}(n)$*

$$DS_{1,q}(n) \leqslant \frac{3 \cdot q^{n-1}}{(n-5)(n-3)(q-1)} + 5q$$

Proof: Note that the number of words in Σ_q^{n-1} with r runs is given by $q(q-1)^{r-1}\binom{n-2}{r-1}$ [11]. The sum over all words in Σ_q^{n-1} using the indicated fractional transversals $w_{\boldsymbol{y}}^1$ has to be computed. For $r = 1, 2, 3$ define the function $g(q,n) = \sum_{r=1}^{3} q(q-1)^{r-1}\binom{n-2}{r-1} w_{\boldsymbol{y}}^1$. The rest is given by

$$\sum_{r=4}^{n-1} q(q-1)^{r-1}\binom{n-2}{r-1} \frac{1}{(r-2)\left[(n-3)(q-1)+(q-2)\right]+(q+2)}$$
$$\leqslant \frac{q}{(n-3)(q-1)} \sum_{r=4}^{n-1} (q-1)^{r-1} \binom{n-2}{r-1} \frac{1}{r-2}.$$

For simplicity, first the following analysis is performed.

$$f(q,n) := \sum_{r=4}^{n-1} (q-1)^{r-1} \binom{n-2}{r-1} \frac{1}{r-2}$$
$$= \sum_{r=2}^{n-3} (q-1)^{r+1} \frac{(n-2)!}{(n-r-3)!\, r!} \left(\frac{1}{r} - \frac{1}{r+1}\right)$$
$$= (n-2)(q-1) \sum_{r=2}^{n-3} (q-1)^r \binom{n-3}{r} \frac{1}{r} - \sum_{r=2}^{n-3} (q-1)^{r+1} \binom{n-2}{r+1}.$$

In the last expression the right part of the difference can be calculated as follows

$$\sum_{r=2}^{n-3} (q-1)^{r+1} \binom{n-2}{r+1} = \sum_{r=3}^{n-2} (q-1)^r \binom{n-2}{r}$$
$$= q^{n-2} - \frac{(q-1)^2(n-2)(n-3)}{2} - (q-1)(n-2) - 1.$$

For the left part, Claim 2 can be used to derive the following inequality

$$(n-2)(q-1) \sum_{r=2}^{n-3} (q-1)^r \binom{n-3}{r} \frac{1}{r}$$
$$\leqslant (n-2)(q-1) \left[\frac{q^{n-2}}{(q-1)(n-5)} - (n-3)(q-1)\right].$$

Thus, an upper bound for $f(q,n)$ can be derived by

$$f(q,n) \leqslant \left[\frac{(n-2)}{(n-5)} - 1\right] q^{n-2}$$
$$- \frac{1}{2}(n-2)(n-3)(q-1)^2 + (q-1)(n-2) + 1.$$

Next, the computed $f(q,n)$ and $g(q,n)$ are combined in the following manner

$$DS_{1,q}(n) \leqslant \frac{q \cdot f(q,n)}{(n-3)(q-1)} + g(q,n).$$

Finally, the bound in the theorem results after some basic algebraic steps and the fact that $q \leqslant n$. ∎

The last theorem provides the following corollary.

Corollary 5. *It holds that* $DS_{1,q}(n) \lesssim \frac{3\cdot q^{n-1}}{n^2(q-1)}$.

B. Upper Bound on Single-Deletion s-Substitution Codes

To state a legitimate fractional transversal for the case of s substitutions, first a lower bound on the cardinality of the ball size $|B^{DS}_{1,s}(\boldsymbol{x})|$ has to be derived. In the remaining part of the section only Σ_2 will be considered.

Claim 3. *For all* $\boldsymbol{x} \in \Sigma_2^n$, *it holds that* $|B^{DS}_{1,s}(\boldsymbol{x})| \geqslant r(\boldsymbol{x})\binom{n-1-s}{s}$.

Note that this lower bound is derived based upon an explicit expression of $|B^{DS}_{1,s}(\boldsymbol{x})|$ from [15]. Using this result, a fractional transversal for the single-deletion s-substitution case can be formulated.

Lemma 6. *The following choice of* $w_{\boldsymbol{y}}$ *with* $\boldsymbol{y} \in \Sigma_2^{n-1}$ *and* $n \geqslant 2s+1 \geqslant 3$ *is a fractional transversal for* $\mathcal{H}_s$

$$w^s_{\boldsymbol{y}} = \begin{cases} \frac{1}{\binom{n-s-1}{s}} & r(\boldsymbol{y}) \leqslant 2s+1, \\ \frac{1}{(r(\boldsymbol{y})-2s)\binom{n-s-1}{s}} & r(\boldsymbol{y}) > 2s+1. \end{cases}$$

As a result of Lemma 6 an upper bound for the cardinality of a single-deletion s-substitution correcting code can be stated.

Theorem 7. *For* $n \geqslant 3$ *the following is an upper bound on* $DS_{s,2}(n)$*:*

$$DS_{s,2}(n) \leqslant \frac{s!(2s+1)}{(n-2s)^s(n-1)}\left[2^n + \frac{2(n-1)^{2s+1}}{2s+1}\right].$$

Proof: First, only the words in $\boldsymbol{y} \in \Sigma_2^{n-1}$ with $r(\boldsymbol{y}) \leqslant 2s+1$ are considered. Using the inequalities are $\sum_{i=0}^{k}\binom{n}{i} \leqslant \sum_{i=0}^{k} n^i \cdot 1^{k-i} \leqslant (1+n)^k$ and $\binom{n}{k} \geqslant \frac{(n-k+1)^k}{k!}$, the sum can be calculated as follows

$$\sum_{r=1}^{2s+1} 2\binom{n-2}{r-1}\frac{1}{\binom{n-s-1}{s}} = \frac{2}{\binom{n-s-1}{s}}\sum_{r=0}^{2s}\binom{n-2}{r} \leqslant \frac{2s!(n-1)^{2s}}{(n-2s)^s}.$$

In a next step, by additionally applying the inequality $\frac{1}{r-2s} \leqslant \frac{2s+1}{r}$ the sum for all words with $2s+2 \leqslant r(\boldsymbol{y}) \leqslant n-1$ can be computed as

$$\sum_{r=2s+2}^{n-1} 2\binom{n-2}{r-1}\frac{1}{\binom{n-s-1}{s}}\frac{1}{r-2s} \leqslant \frac{2}{\binom{n-s-1}{s}}\sum_{r=2s+2}^{n-1}\binom{n-2}{r-1}\frac{2s+1}{r}$$
$$= \frac{2s+1}{n-1}\frac{2}{\binom{n-s-1}{s}}\sum_{r=2s+2}^{n-1}\binom{n-1}{r} \leqslant \frac{2s+1}{n-1}\frac{2s!}{(n-2s)^s}\cdot 2^{n-1}.$$

Subsequently, the sum of all words with $r \leqslant 2s+1$ is added to the equation again which results to the expression in the theorem. ∎

The corollary below concludes the previous result.

Corollary 8. *It holds that* $DS_{s,2}(n) \lesssim \frac{s!(2s+1)\cdot 2^n}{n^{s+1}}$.

Note that unlike the proof of Theorem 4, in the proof of Theorem 7 Claim 2 is not applied. Instead, Claim 3 and a different upper bound is used. For this reason, in Corollary 8 the bound for the value of $s = 1$ is not the same as the bound stated in Corollary 5 with $q = 2$.

IV. Properties of Codes

In this section, several properties of the families of codes studied in the paper are presented. Consider a family of codes which are defined in the following way. A binary code $\mathcal{C} \subseteq \Sigma_2^n$ is called a $(\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n); a, N)$*-congruent code* if it is defined in the following way

$$\mathcal{C}(\boldsymbol{\gamma}; a, N) = \left\{\boldsymbol{c} \in \Sigma_2^n \;\middle|\; \sum_{i=1}^{n}\gamma_i \cdot c_i \equiv a \pmod N\right\}.$$

The lemmas in this section provide several basic properties in case the intersection of the single-deletion single-substitution balls of two codewords in $\mathcal{C}$ is not trivial. For the rest of this section it is assumed that $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}(\boldsymbol{\gamma}; a, N)$, where $B^{DS}_{1,1}(\boldsymbol{x}) \cap B^{DS}_{1,1}(\boldsymbol{y}) \neq \emptyset$ so that there exists $\boldsymbol{z} \in B^{DS}_{1,1}(\boldsymbol{x}) \cap B^{DS}_{1,1}(\boldsymbol{y})$. For simplicity of notation, the expression $\boldsymbol{x}(d,e)$ is defined to be the error-word achieved from $\boldsymbol{x}$ by deleting the bit in the index d, and substituting the bit in the index e. The variables $d_x, d_y, e_x, e_y \in [n]$ are indices such that $\boldsymbol{z} = \boldsymbol{x}(d_x, e_x) = \boldsymbol{y}(d_y, e_y)$. It can be assumed w.l.o.g. that $d_x < d_y$. In order to shift the values of the substituted bits from binary to ± 1, the following notation is used $\delta_i := 2 \cdot x_i - 1$.

Lemma 9. *For* $e_x, e_y \notin [d_x, d_y]$ *the following statements hold.*

1) *For* $i \in [d_x+1, d_y]$, $x_i = y_{i-1}$.
2) *For* $i \in \{e_x, e_y\}$, $x_i = \overline{y_i} = 1 - y_i$.
3) *For* $i \in [n] \setminus ([d_x, d_y] \cup \{e_x, e_y\})$, $x_i = y_i$.
4) $\sum_{i=d_x}^{d_y}\gamma_i \cdot (x_i - y_i) =$ $x_{d_x}\cdot\gamma_{d_x} + \sum_{i=d_x+1}^{d_y}(\gamma_i - \gamma_{i-1})\cdot x_i - y_{d_y}\cdot\gamma_{d_y}$
5) $\delta_{e_x}\cdot\gamma_{e_x} + \delta_{e_y}\cdot\gamma_{e_y} + x_{d_x}\cdot\gamma_{d_x} - y_{d_y}\cdot\gamma_{d_y}$ $+ \sum_{i=d_x+1}^{d_y} x_i \cdot(\gamma_i - \gamma_{i-1}) \equiv 0 \pmod N$

Proof: For any $i \in [d_x+1, d_y]$ the definition of $\boldsymbol{z}$ leads to the fact that $z_i = y_i$ and also $z_i = x_{i+1}$. This proves statement 1, and a similar proof can be shown for statement 3.

For a substituted bit $i = e_x$ either $i < d_x$ in which case, $x_i = \overline{z_i}, y_i = z_i$. The cases of $i = e_y$ and $i > d_y$ can be proved in a similar way. This concludes the proof of statement 2.

In order to prove statement 4, the sum is separated in the following manner

$$\sum_{i=d_x}^{d_y}\gamma_i\cdot(x_i - y_i) = x_{d_x}\cdot\gamma_{d_x} - y_{d_y}\cdot\gamma_{d_y} + \sum_{i=d_x+1}^{d_y}\gamma_i\cdot x_i - \sum_{i=d_x}^{d_y-1}\gamma_i\cdot y_i.$$

Using statement 1, the last element is simplified as follows

$$\sum_{i=d_x}^{d_y-1}\gamma_i\cdot y_i = \sum_{i=d_x+1}^{d_y}\gamma_{i-1}\cdot y_{i-1} = \sum_{i=d_x+1}^{d_y}\gamma_{i-1}\cdot x_i.$$

Thus, the equality can be rewritten as

$$x_{d_x}\cdot\gamma_{d_x} - y_{d_y}\cdot\gamma_{d_y} + \sum_{i=d_x+1}^{d_y}\gamma_i\cdot x_i - \sum_{i=d_x}^{d_y-1}\gamma_i\cdot y_i$$
$$= x_{d_x}\cdot\gamma_{d_x} - y_{d_y}\cdot\gamma_{d_y} + \sum_{i=d_x+1}^{d_y}(\gamma_i - \gamma_{i-1})\cdot x_i.$$

This concludes the proof of statement 4.

The fact $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}(\boldsymbol{\gamma}; a, N)$ implies that $\sum_{i=1}^{n}\gamma_i x_i \equiv \sum_{i=1}^{n}\gamma_i y_i \equiv a \pmod N$. From this follows

$$\sum_{i=1}^{n}\gamma_i x_i - \sum_{i=1}^{n}\gamma_i y_i \equiv 0 \pmod N.$$

Furthermore, Statement 3 implicates that $\sum_{i=1}^{d_x-1}\gamma_i\cdot(x_i-y_i)+\sum_{i=d_y+1}^{n}\gamma_i\cdot(x_i-y_i)=\gamma_{e_x}\cdot(x_{e_x}-y_{e_x})+\gamma_{e_y}\cdot(x_{e_y}-y_{e_y})$. On the other hand statement 2 leads to the fact that for $e\in\{e_x,e_y\}$: $x_e-y_e=x_e-\overline{x_e}=2\cdot x_e-1=\delta_{x_e}$. Combined together with statement 4 the following equivalence is achieved

$$\delta_{e_x}\cdot\gamma_{e_x}+\delta_{e_y}\cdot\gamma_{e_y}+x_{d_x}\cdot\gamma_{d_x}+\sum_{i=d_x+1}^{d_y}x_i\cdot(\gamma_i-\gamma_{i-1})-y_{d_y}\cdot\gamma_{d_y}$$
$$=\sum_{i=1}^{n}\gamma_i x_i-\sum_{i=1}^{n}\gamma_i y_i\equiv 0 \pmod N.$$

This proves statement 5. ■

In a similar manner, the following lemma can be proved.

Lemma 10. *The following conditions hold:*

1) *For $e_x\in[d_x,d_y]$ and $e_y\notin[d_x,d_y]$*
$$\delta_{e_x}\cdot\gamma_{e_x-1}+\delta_{e_y}\cdot\gamma_{e_y}+x_{d_x}\cdot\gamma_{d_x}-y_{d_y}\cdot\gamma_{d_y}+ +\textstyle\sum_{i=d_x+1}^{d_y}x_i\cdot(\gamma_i-\gamma_{i-1})\equiv 0 \pmod N$$
2) *For $e_x,e_y\in[d_x,d_y]$*
$$\delta_{e_x}\cdot\gamma_{e_x-1}+\delta_{e_y+1}\cdot\gamma_{e_y}+x_{d_x}\cdot\gamma_{d_x}-y_{d_y}\cdot\gamma_{d_y}+ +\textstyle\sum_{i=d_x+1}^{d_y}x_i\cdot(\gamma_i-\gamma_{i-1})\equiv 0 \pmod N$$

Define the variable ϵ_x as 1 if $e_x\in[d_x,d_y]$ and 0 otherwise. The variable ϵ_y is defined in a similar manner. An important claim about the property of the code is as follows.

Claim 4. *For any $\boldsymbol{x},\boldsymbol{y}\in\Sigma_2^n$*

1) *$\boldsymbol{x}(d_x,e_x)=\boldsymbol{y}(d_y,e_y)$ iff $\boldsymbol{x}(d_x,e_y+\epsilon_y)=\boldsymbol{y}(d_y,e_x-\epsilon_x)$.*
2) *$B_{1,1}(\boldsymbol{x})\cap B_{1,1}(\boldsymbol{y})\neq\emptyset$ iff $B_{1,1}(\overline{\boldsymbol{x}})\cap B_{1,1}(\overline{\boldsymbol{y}})\neq\emptyset$.*
3) *For some weight vector $\boldsymbol{\gamma}\in\mathbb{Z}^n$, a,N, and for any $\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}(\boldsymbol{\gamma};a,N)$ there exists such b so that $\overline{\boldsymbol{x}},\overline{\boldsymbol{y}}\in\mathcal{C}(\boldsymbol{\gamma};b,N)$.*

V. Construction

In this section, the main result of the paper is shown. An explicit construction for a single-deletion single-substitution correcting code and its correctness are presented. This construction requires redundancy of at most $6\cdot\log(n)+8$ bits.

Construction 11. *Define four weight vectors $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta},\mathbb{1}\in\mathbb{Z}^n$ in the following way*

$$\boldsymbol{\alpha}=(1,2,3,\ldots,n),\boldsymbol{\beta}=(\sum_{i=1}^{1}i,\sum_{i=1}^{2}i,\sum_{i=1}^{3}i,\ldots,\sum_{i=1}^{n}i),$$

$$\boldsymbol{\eta}=(\sum_{i=1}^{1}i^2,\sum_{i=1}^{2}i^2,\sum_{i=1}^{3}i^2,\ldots,\sum_{i=1}^{n}i^2),\mathbb{1}=(1,\ldots,1).$$

For fixed integers $a\in[3n],b\in[3\cdot n^2],c\in[3\cdot n^3],d\in[4]$, the code is defined as

$$\mathcal{C}_{a,b,c,d}=\mathcal{C}(\boldsymbol{\alpha};a,3n+1)\cap\mathcal{C}(\boldsymbol{\beta};b,3n^2+1)$$
$$\cap\;\mathcal{C}(\boldsymbol{\eta};c,3n^3+1)\cap\mathcal{C}(\mathbb{1};d,5)$$

Remember that d_x,d_y are denoted as the indices of the deleted bits from $\boldsymbol{x},\boldsymbol{y}$ respectively. The definition of e_x,e_y are the indices of the substituted bits from $\boldsymbol{x},\boldsymbol{y}$. The definition of ϵ_x is 1 if $e_x\in[d_x,d_y]$ and 0 otherwise. The definition of ϵ_y is similar.

Theorem 12. *For any indices $e_x,e_y,d_x,d_y\in[n]$, any two codewords $\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}_{a,b,c,d}$ fulfill*

$$\boldsymbol{x}(d_x,e_x)\neq\boldsymbol{y}(d_y,e_y).$$

Proof: For simplicity, denote $\widehat{e}_x=e_x-\epsilon_x$ and $\widehat{e}_y=e_y+\epsilon_y$. Assume by contradiction that $\boldsymbol{x}(d_x,e_x)=\boldsymbol{y}(d_y,e_y)$. The following equivalence can be concluded from Lemma 9 (statement 5), Lemma 10, and the fact that $\mathcal{C}_{a,b,c,d}\subseteq\mathcal{C}(\mathbb{1};d,5)$.

$$\delta_{e_x}+\delta_{\widehat{e}_y}+x_{d_x}-y_{d_y}+0\equiv 0 \bmod 5$$

This is equivalent to the following system of equations

$$x_{d_x}=y_{d_y},\ \delta_{e_x}=-\delta_{\widehat{e}_y}.$$

Define the set $\mathcal{S}$ as follows.

$$\mathcal{S}:=\{d_x+1\leqslant i\leqslant d_y|x_i=1\}\subseteq[d_x,d_y].$$

According to Claim 4 statement 1, it is possible to assume w.l.o.g. that $e_x<e_y$. According to Claim 4 statements 2-3, it is also possible to assume w.l.o.g. that $x_{d_x}=y_{d_y}=0$. By substituting these values into Lemma 9 statement 5 and Lemma 10 the following equivalence is achieved.

For any $k\in\{0,1,2\}$

$$\delta_{e_x}\cdot\sum_{j=1}^{\widehat{e}_x}j^k+\delta_{\widehat{e}_y}\cdot\sum_{j=1}^{e_y}j^k+\sum_{j\in\mathcal{S}}j^k\equiv 0 \bmod 3n^{k+1}+1.$$

Notice that $|\delta_{e_x}\cdot\sum_{j=1}^{\widehat{e}_x}j^k|<\sum_{j=1}^{n}j^k<n\cdot n^k=n^{k+1}$. This is also true for $|\delta_{\widehat{e}_y}\cdot\sum_{j=1}^{e_y}j^k|$ and $|\sum_{j\in\mathcal{S}}j^k|$. As a result, the left part of the equivalences is at least $-3\cdot n^{k+1}$ and at most $3\cdot n^{k+1}$. Therefore, the congruences are strict equalities.

The equalities can be rewritten as

$$\sum_{j\in\mathcal{S}}j^k=-\delta_{\widehat{e}_y}\cdot(-\sum_{j=1}^{\widehat{e}_x}j^k+\sum_{j=1}^{e_y}j^k)=-\delta_{\widehat{e}_y}\cdot(\sum_{\widehat{e}_x+1}^{e_y}j^k).$$

Notice that the left hand side is always non-negative. Hence, the sign of the right hand side is non-negative as well. From this follows that $\delta_{\widehat{e}_y}=-1$ and the equation can be transformed to

$$\sum_{\widehat{e}_x+1}^{e_y}j^k=\sum_{j\in\mathcal{S}}j^k. \tag{1}$$

Since the current assumption is that $d_x<d_y$ and $\widehat{e}_x<e_y$, there are 6 possible orderings of the 4 indices. A full proof for the cases $\widehat{e}_x,e_y<d_x$, $\widehat{e}_x<d_x<e_y<d_y$ will follow, and a guidance for the rest of the cases can be found afterwards.

Assume $\widehat{e}_x,e_y<d_x$. In this case, two sets are defined as

$$\mathcal{S}_1:=[\widehat{e}_x+1,e_y],\ \mathcal{S}_2:=\mathcal{S}.$$

Notice that for any two indices $i\in\mathcal{S}_1,\ j\in\mathcal{S}_2$ the following holds

$$i<j. \tag{2}$$

Hence, (1) can be altered to the following form. For any $k\in\{0,1,2\}$

$$\sum_{j\in\mathcal{S}_1}j^k=\sum_{j\in\mathcal{S}_2}j^k.$$

For $k = 0$ this equality is $|\mathcal{S}_1| = |\mathcal{S}_2|$ which means the cardinality of the sets is equal. For $k = 1$ this equality is $\sum_{j\in\mathcal{S}_1} j = \sum_{j\in\mathcal{S}_2} j$ which means the sum of elements of $\mathcal{S}_1, \mathcal{S}_2$ is equal as well. However, through equality (2) if the cardinality of the sets is the same then the sum of elements in $\mathcal{S}_2$ should be strictly bigger than the sum of elements in $\mathcal{S}_1$. This concludes this case.

Assume $\widehat{e}_x < d_x < e_y < d_y$. In this case, three sets are defined as

$$\mathcal{S}_1 := [\widehat{e}_x + 1, d_x],\ \mathcal{S}_2 := [d_x + 1, e_y] \setminus \mathcal{S},\ \mathcal{S}_3 := [e_y + 1, d_y] \cap \mathcal{S}$$

Observe that for any three indices $i \in \mathcal{S}_1,\ j \in \mathcal{S}_2,\ \ell \in \mathcal{S}_3$ the following holds

$$i < j < \ell. \tag{3}$$

Now, (1) can be rewritten as follows. For any $k \in \{0, 1, 2\}$

$$\sum_{j\in\mathcal{S}_1} j^k + \sum_{j\in\mathcal{S}_2} j^k - \sum_{j\in\mathcal{S}_3} j^k = 0.$$

This can be written in matrix form. There exist integers v_1, v_2, v_3 such that at least one of them is non-zero and the following equality holds

$$\mathbf{A} \cdot \boldsymbol{v} := \begin{pmatrix} \sum_{j\in\mathcal{S}_1} 1 & \sum_{j\in\mathcal{S}_2} 1 & \sum_{j\in\mathcal{S}_3} 1 \\ \sum_{j\in\mathcal{S}_1} j & \sum_{j\in\mathcal{S}_2} j & \sum_{j\in\mathcal{S}_3} j \\ \sum_{j\in\mathcal{S}_1} j^2 & \sum_{j\in\mathcal{S}_2} j^2 & \sum_{j\in\mathcal{S}_3} j^2 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

In this case, $v_1 = v_2 = 1, v_3 = -1$ is such a solution. This equality means $\mathbf{A}$ has a non-trivial solution to the homogeneous system of equalities, which also means $\det(\mathbf{A}) = 0$.

The determinant of the matrix $\mathbf{A}$ can be computed by

$$\begin{vmatrix} \sum_{j\in\mathcal{S}_1} 1 & \sum_{j\in\mathcal{S}_2} 1 & \sum_{j\in\mathcal{S}_3} 1 \\ \sum_{j\in\mathcal{S}_1} j & \sum_{j\in\mathcal{S}_2} j & \sum_{j\in\mathcal{S}_3} j \\ \sum_{j\in\mathcal{S}_1} j^2 & \sum_{j\in\mathcal{S}_2} j^2 & \sum_{j\in\mathcal{S}_3} j^2 \end{vmatrix} = \sum_{i\in\mathcal{S}_1} \sum_{j\in\mathcal{S}_2} \sum_{k\in\mathcal{S}_3} \begin{vmatrix} 1 & 1 & 1 \\ i & j & k \\ i^2 & j^2 & k^2 \end{vmatrix}.$$

Notice that each element in the sum is a determinant of a Vandermonde matrix. Hence,

$$\begin{vmatrix} 1 & 1 & 1 \\ i & j & k \\ i^2 & j^2 & k^2 \end{vmatrix} = (j - i) \cdot (k - j) \cdot (k - i).$$

To summarize, (3) the following contradiction is obtained.

$$\begin{aligned} 0 &= \sum_{i\in\mathcal{S}_1} \sum_{j\in\mathcal{S}_2} \sum_{k\in\mathcal{S}_3} \begin{vmatrix} 1 & 1 & 1 \\ i & j & k \\ i^2 & j^2 & k^2 \end{vmatrix} \\ &= \sum_{i\in\mathcal{S}_1} \sum_{j\in\mathcal{S}_2} \sum_{k\in\mathcal{S}_3} (j - i) \cdot (k - j) \cdot (k - i) > 0. \end{aligned}$$

This concludes this case.

For any of the other orderings, the same proof can be concluded using the following definitions:

1) For $\widehat{e}_x, e_y < d_x$, define $\mathcal{S}_1 := [\widehat{e}_x + 1, e_y], \mathcal{S}_2 := \mathcal{S}$;
2) For $\widehat{e}_x < d_x < e_y < d_y$, define $\mathcal{S}_1 := [\widehat{e}_x + 1, d_x], \mathcal{S}_2 := [d_x + 1, e_y] \setminus \mathcal{S}, \mathcal{S}_3 := [e_y + 1, d_y] \cap \mathcal{S}$;
3) For $\widehat{e}_x < d_x < d_y < e_y$, define $\mathcal{S}_1 := [\widehat{e}_x + 1, d_x], \mathcal{S}_2 := \mathcal{S}, \mathcal{S}_3 := [d_y + 1, e_y]$;
4) For $d_x < \widehat{e}_x < e_y < d_y$, define $\mathcal{S}_1 := \mathcal{S} \cap [d_x, \widehat{e}_x], \mathcal{S}_2 := [\widehat{e}_x + 1, e_y] \setminus \mathcal{S}, \mathcal{S}_3 := [e_y + 1, d_y] \cap \mathcal{S}$;
5) For $d_x < \widehat{e}_x < d_y < e_y$, define $\mathcal{S}_1 := \mathcal{S} \cap [d_x, \widehat{e}_x], \mathcal{S}_2 := [\widehat{e}_x + 1, e_y] \setminus \mathcal{S}, \mathcal{S}_3 := [d_y + 1, e_y]$;
6) For $d_y < \widehat{e}_x, e_y$, define $\mathcal{S}_1 := \mathcal{S}, \mathcal{S}_2 := [\widehat{e}_x + 1, e_y]$.

This concludes the proof. ■

In this proof, it is shown that $\mathcal{C}_{a,b,c,d}$ guarantees to correct a combination of single-deletion and single-substitution errors. However, a single deletion or a single substitution can be corrected as well due to the $\boldsymbol{\alpha}$ constraint of the code $\mathcal{C}(\boldsymbol{\alpha}; a, 3n + 1)$ [6]. Lastly, via the pigeonhole principle the following conclusion about the redundancy is obtained.

Corollary 13 *There exist $a \in [3n + 1], b \in [3n^2 + 1], c \in [3n^3 + 1], d \in [5]$ such that the code $\mathcal{C}_{a,b,c,d}$ is a binary single-deletion single-substitution correcting code with at most $6 \cdot \log(n) + 8$ redundancy bits.*

VI. Construction for Non-Binary Alphabets

In this section, a construction for non-binary codes correcting a single deletion and a single substitution is presented.

For a word $\boldsymbol{z} \in \Sigma_q^n$ we associate its *binary signature* $\boldsymbol{z}^{01} \in \Sigma_2^n$, where $z_1^{01} = 1$ and $z_i^{01} = 1$ if and only if $z_i > z_{i-1}$. The motivation for using the signature vector is to convert the error correction into a binary problem. The following lemma shows the conversion explicitly. We define an *adjacent transposition* to be the error event in which two adjacent bits switch their positions.

Lemma 14. *For any word $\boldsymbol{x} \in \Sigma_q^n$, and $\boldsymbol{y} \in \Sigma_q^{n-1}$ the error word achieved by a single-deletion and a single-substitution, $\boldsymbol{y}^{01}$ can be achieved from by $\boldsymbol{x}^{01}$ by one of the following errors:*

1) *single-deletion;*
2) *single-deletion and a single-substitution;*
3) *single-deletion and a single-adjacent-transposition.*

For the rest of this section, let $\mathcal{C}_2$ be a code correcting either a single deletion and a single substitution or a single deletion and a single adjacent transposition. We are now ready to present the code construction for the non-binary case.

Construction 15. *For $a \in [2q], b \in [2qn], c \in [2qn^2]$, let $\mathcal{C}_{a,b,c}$ be the code*

$$\mathcal{C}_{a,b,c} = \left\{ \boldsymbol{x} \in \Sigma_q^n | \boldsymbol{x}^{01} \in \mathcal{C}_2,\ \begin{aligned} \textstyle\sum_{j=1}^n x_j &\equiv a (\bmod\ 2 \cdot q + 1), \\ \textstyle\sum_{j=1}^n j \cdot x_j &\equiv b (\bmod\ 2 \cdot n \cdot q + 1), \\ \textstyle\sum_{j=1}^n j^2 \cdot x_j &\equiv c (\bmod\ 2 \cdot n^2 \cdot q + 1) \end{aligned} \right\}$$

The next theorem states the correctness of this code construction.

Theorem 16. *For all $a \in [2q], b \in [2qn], c \in [2qn^2]$ the code $\mathcal{C}_{a,b,c} \subseteq \Sigma_q^n$ is a single-deletion single-substitution correcting code.*

Notice that this construction requires an extension of the binary code presented in Section V. Such an extension is possible using the result presented in [16] for a family of binary codes correcting a single deletion and a single adjacent-transposition. Hence, it is possible to conclude with the following corollary.

Corollary 17. *There exist $a \in [2q], b \in [2qn], c \in [2qn^2]$ such that the code $\mathcal{C}_{a,b,c}$ is a single-deletion single-substitution code with at most $10 \cdot \log(n) + 3 \cdot \log(q) + 11$ redundancy symbols.*

References

[1] R. Heckel, G. Mikutis, and R. N. Grass, "A Characterization of the DNA Data Storage Channel," *Scientific Reports*, vol. 9, no. 1, p. 9663, 2019. [Online]. Available: https://doi.org/10.1038/s41598-019-45832-6

[2] F. Sala, C. Schoeny, N. Bitouz, and L. Dolecek, "Synchronizing files from a large number of insertions and deletions," *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2258–2273, June 2016.

[3] L. Dolecek and V. Anantharam, "Using ReedMuller RM (1, m) codes over channels with synchronization and substitution errors," *IEEE Transactions on Information Theory*, vol. 53, no. 4, pp. 1430–1443, April 2007.

[4] G. Tenengolts, "Nonbinary codes, correcting single deletion or insertion (corresp.)," *IEEE Trans. Inf. Theory*, vol. 30, no. 5, pp. 766–769, 1984.

[5] R. R. Varshamov and G. M. Tenengolts, "Codes which correct single asymmetric errors (in Russian)," *Automatika i Telemkhanika*, vol. 161, no. 3, pp. 288–292, 1965.

[6] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals (in Russian)," *Doklady Akademii Nauk SSR*, vol. 163, no. 4, pp. 845–848, 1965.

[7] J. Brakensiek, V. Guruswami, and S. Zbarsky, "Efficient low-redundancy codes for correcting multiple deletions," *IEEE Trans. Inform. Theory*, 2017.

[8] R. Gabrys and F. Sala, "Codes correcting two deletions," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 965–974, Feb 2019.

[9] J. Sima, N. Raviv, and J. Bruck, "Two deletion correcting codes from indicator vectors," *IEEE Transactions on Information Theory*, pp. 1–1, 2019.

[10] J. Sima and J. Bruck, "Optimal k-deletion correcting codes," in *2019 IEEE International Symposium on Information Theory (ISIT)*, July 2019, pp. 847–851.

[11] A. A. Kulkarni and N. Kiyavash, "Nonasymptotic upper bounds for deletion correcting codes," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 5115–5130, Aug 2013.

[12] A. Fazeli, A. Vardy, and E. Yaakobi, "Generalized sphere packing bound," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2313–2334, May 2015.

[13] M. Abu-Sini and E. Yaakobi, "Reconstruction of sequences in DNA storage," in *2019 IEEE International Symposium on Information Theory (ISIT)*, July 2019, pp. 290–294.

[14] R. Gabrys, E. Yaakobi, and L. Dolecek, "Correcting grain-errors in magnetic media," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2256–2272, May 2015.

[15] M. Abu-Sini and E. Yaakobi, "Reconstruction of sequences in DNA storage," to be submitted *IEEE Transactions on Information Theory*.

[16] R. Gabrys, E. Yaakobi, and O. Milenkovic, "Codes in the damerau distance for deletion and adjacent transposition correction," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2550–2570, April 2018.