Codes for Correcting Tandem Repeats

Yeow Meng Chee, Johan Chrisnata, Han Mao Kiah, and Tuan Thanh Nguyen
School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
email: {ymchee, jchrisnata, hmkiah, nguyentu001}@ntu.edu.sg

***Abstract*—Tandem duplication in DNA is the process of inserting a copy of a segment of DNA adjacent to the original position. Motivated by applications that store data in living organisms, Jain *et al.* (2016) proposed the study of codes that correct tandem duplications to improve the reliability of data storage. We investigate algorithms associated with the study of these codes.**

Two words are said to be $\leqslant k$-confusable if there exists two sequences of tandem duplications of lengths at most k such that the resulting words are equal. We demonstrate that the problem of deciding whether two words is $\leqslant k$-confusable is linear-time solvable through a characterisation that can be checked efficiently for $k = 3$.

Using insights gained from the algorithm, we study the size of tandem-duplication codes. We improve the previous known upper bound and then construct codes with larger sizes as compared to the previous constructions. We determine the sizes of optimal tandem-duplication codes for lengths up to twenty, and develop recursive methods to construct tandem-duplication codes for all word lengths.

I. Introduction

Lander *et al.* [1] published a draft sequence of the human genome and reported that more than 50% of the genome consists of repeated substrings. There are two types of common repeats: *interspersed* and *tandem* repeats. Interspersed repeats are caused by transposons when a segment of DNA is copied and pasted into new positions of the genome. In contrast, tandem repeats are caused by slipped-strand mispairings [2], and they occur when a pattern of one or more nucleotides is repeated and the repetitions are adjacent to each other. For example, consider the word AGTAGTCTGC. The substring AGTAGT is a tandem repeat, and we say that AGTAGTCTGC is generated from AGTCTGC by a *tandem duplication* of length three. Tandem repeats are believed to be the cause of several genetic disorders [3].

Recently, motivated by applications that store data in living organisms [4]–[6], Jain *et al.* [7] proposed the study of codes that correct tandem duplications to improve the reliability of data storage. They investigated various types of tandem duplications and provided optimal code construction in the case where duplication length is at most two.

In this work, we investigate algorithms associated with these codes. In particular, given two words $\mathbf{x}$ and $\mathbf{y}$, we look for efficient algorithms that answer the following question: When are the words $\mathbf{x}$ and $\mathbf{y}$ *confusable* under tandem repeats? In the full version of this paper [8], we derive sufficient and necessary conditions for two strings to be confusable and propose a linear-time algorithm to determine the confusability of any two strings assuming that the length of tandem repeats is at most three.

Due to space constraints, we omit the description of the algorithm here. Instead, we summarise the results on the code sizes that were obtained via insights from the algorithm.

II. Notations and Terminology

Let $\Sigma_q = \{0, 1, \cdots, q-1\}$. Let Σ_q^n denote the set of all sequences of length n over Σ_q, and let Σ_q^* denote the set of all finite sequences over Σ_q. Given two words $\boldsymbol{x}, \boldsymbol{y} \in \Sigma_q^*$, we denote their concatenation by $\boldsymbol{xy}$.

We state the *tandem duplication* rules. For nonnegative integers $k \leqslant n$ and $i \leqslant n-k$, we define $T_{i,k} : \Sigma_q^n \to \Sigma_q^{n+k}$ such that $T_{i,k}(\boldsymbol{x}) = \boldsymbol{uvvw}$, where $\boldsymbol{x} = \boldsymbol{uvw}$, $|\boldsymbol{u}| = i$, $|\boldsymbol{v}| = k$.

If a finite sequence of tandem duplications of *length at most* k is performed to obtain $\boldsymbol{y}$ from $\boldsymbol{x}$, then we say that $\boldsymbol{y}$ is a $\leqslant k$-*descendant* of $\boldsymbol{x}$, or $\boldsymbol{x}$ is a $\leqslant k$-*ancestor* of $\boldsymbol{y}$, and denote this relation by $\boldsymbol{x} \underset{\leqslant k}{\overset{*}{\Longrightarrow}} \boldsymbol{y}$. We define the $\leqslant k$-*descendant cone* of $\boldsymbol{x}$ to be the set of all $\leqslant k$-descendants of $\boldsymbol{x}$ and denote this cone by $D^*_{\leqslant k}(\boldsymbol{x})$.

Motivated by applications that store data on living organisms, Jain *et al.* [7] looked at the $\leqslant k$-descendant cones of a pair of words and asked whether the two cones have a nonempty intersection. Specifically, we introduce the notion of confusability.

Definition 1 (Confusability)**.** Two words $\boldsymbol{x}$ and $\boldsymbol{y}$, are said to be $\leqslant k$-*confusable* if $D^*_{\leqslant k}(\boldsymbol{x}) \cap D^*_{\leqslant k}(\boldsymbol{y}) \neq \varnothing$.

To design error-correcting codes that store information in the DNA of living organisms, Jain *et al.* then proposed the use of codewords that are not pairwise confusable.

Definition 2 ($\leqslant k$-Tandem-Duplication Codes)**.** A subset $\mathcal{C} \subseteq \Sigma_q^n$ is a $\leqslant k$-*tandem-duplication code* if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$ and $\boldsymbol{x} \neq \boldsymbol{y}$, we have that $\boldsymbol{x}$ and $\boldsymbol{y}$ are not $\leqslant k$-confusable. We say that $\mathcal{C}$ is an $(n, \leqslant k; q)$-tandem-duplication code or $(n, \leqslant k; q)$-TD code.

We are interested in determining the maximum possible size of an $(n, \leqslant k; q)$-TD code, and we denote the quantity by $T(n, k; q)$.

A. Previous Work

We state known lower and upper bounds on the quantity $T(n, k; q)$. Crucial to these bounds is the concept of irreducible words and roots.

Definition 3. A word $\boldsymbol{x}$ is said to be $\leqslant k$-*irreducible* if $\boldsymbol{x}$ cannot be deduplicated into shorter words with deduplication of length at most k. In other words, if $\boldsymbol{y} \underset{\leqslant k}{\overset{*}{\Longrightarrow}} \boldsymbol{x}$, then $\boldsymbol{y} = \boldsymbol{x}$. The set of $\leqslant k$-irreducible q-ary words is denoted by $\mathrm{Irr}_{\leqslant k}(q)$ and those of length n is denoted by $\mathrm{Irr}_{\leqslant k}(n, q)$. The $\leqslant k$-ancestors of $\boldsymbol{x} \in \Sigma_q^*$ that are $\leqslant k$-irreducible are called the $\leqslant k$-*roots* of $\boldsymbol{x}$, denoted by $R_{\leqslant k}(\boldsymbol{x})$.

Jain *et al.* used irreducible words to construct tandem-duplication codes and demonstrate that the construction is optimal for the case $k = 2$.

Construction 1.

(i) $T(n, 2; 3) = \sum_{i=1}^{n} |\mathrm{Irr}_{\leqslant 2}(i, q)|$.
(ii) $T(n, 3; 3) \geqslant \sum_{i=1}^{n} |\mathrm{Irr}_{\leqslant 3}(i, q)|$.

We next look at upper bounds on the size of an optimal $(n, \leqslant 3; q)$-TD code. By definition, an $(n, \leqslant 3; q)$-TD code is also an $(n, \leqslant 2; q)$-TD code. Since an optimal $(n, \leqslant 2; q)$-TD code is provided by Construction 1, we have the following upper bound on the size of an optimal $(n, \leqslant 3; q)$-TD code.

Proposition 1. $T(n, 3; q) \leqslant \sum_{i=1}^{n} |\mathrm{Irr}_{\leqslant 2}(i, q)|$.

TABLE I: Estimates and Exact Values for $T(n, 3; 3)$

n	Constr. 1	This paper	Eq (1)	Prop. 1	n	Constr. 1	This paper	Eq (1)	Prop. 1	n	Constr. 1	This paper	Eq (1)	Prop. 1
1	**3**	**3**	**3**	**3**	11	867	**1221**	1227	1389	21	40587	83619	125001	171933
2	**9**	**9**	**9**	**9**	12	1281	**1887**	1941	2253	22	59493	116145	199467	278199
3	**21**	**21**	**21**	**21**	13	1887	**2913**	3075	3651	23	87201	166761	318621	450141
4	**39**	**39**	**39**	**39**	14	2775	**4527**	4875	5913	24	127809	249159	509457	728349
5	**69**	**69**	**69**	**69**	15	4077	**6969**	7731	9573	25	187323	375129	815361	1178499
6	111	**117**	**117**	**117**	16	5985	**10641**	12267	15495	26	274545	558573	1306107	1906857
7	171	**195**	**195**	**195**	17	8781	**16287**	19479	25077	27	402375	813771	2093967	3085365
8	261	**315**	**315**	321	18	12879	**25005**	30957	40581	28	589719	1164309	3359685	4992231
9	393	**495**	**495**	525	19	18885	**38223**	49245	65667	29	864285	1675935	5394369	8077605
10	585	**777**	**777**	855	20	27687	**57957**	78417	106257	30	1266681	2464419	8667075	13069845

Optimal values of $T(n, 3; 3)$ are highlighted in **bold**.

Proposition 1 implies that Construction 1 is tight for $k = 3$ and $n \leqslant 5$. Using a combinatorial characterization implied by our algorithm, we improve this upper bound for longer lengths.

III. Tandem Duplication Codes

Motivated by the concept of roots, we consider a $\leqslant$3-irreducible word $\boldsymbol{r}$ and we say that a $(n, \leqslant 3; q)$-TD code $\mathcal{C}$ is an $(n, \leqslant 3; \boldsymbol{r})$-TD code if all words in $\mathcal{C}$ belong to $D^*_{\leqslant 3}(\boldsymbol{r})$. Since $\bigcup_{\boldsymbol{r} \in \mathrm{Irr}_{\leqslant 3}(i,q),\, i \leqslant n} \mathcal{C}(n, \boldsymbol{r})$ is an $(n, \leqslant 3; q)$-TD code, we provide estimates on the size of an optimal $(n, \leqslant 3; \boldsymbol{r})$-TD code for a fixed $\boldsymbol{r}$. To simplify our discussion, we focus on the case $q = 3$ and let $T(n)$ and $T(n, \boldsymbol{r})$ to denote the sizes of an optimal $(n, \leqslant 3; 3)$-TD code and an optimal $(n, \leqslant 3; \boldsymbol{r})$-TD code, respectively.

Via our combinatorial characterisation, we determined the exact values of $T(n, \boldsymbol{r})$ in certain cases.

Proposition 2 (Exact Values)**.**

(i) $T(n, \boldsymbol{r}) = 1$ *for* $|\boldsymbol{r}| \leqslant n \leqslant |\boldsymbol{r}| + 2$.

(ii) $T(|\boldsymbol{r}| + 3, \boldsymbol{r}) = 2$.

(iii) *Let* $\boldsymbol{r} \in R \triangleq \{012, 0120, 01201, 1012, 10120, 101201, 0121, 01202, 012010, 10121, 101202, 1012010\}$. *Set*

$$n_2(\boldsymbol{r}) \triangleq \begin{cases} 10, & \text{if } \boldsymbol{r} = 012, \\ 11, & \text{if } \boldsymbol{r} \in \{0121, 1012\}, \\ 12, & \text{if } \boldsymbol{r} \in \{0120, 10121\}, \\ 13, & \text{if } \boldsymbol{r} \in \{01202, 10120\}, \\ 14, & \text{if } \boldsymbol{r} \in \{01201, 101202\}, \\ 15, & \text{if } \boldsymbol{r} \in \{012010, 101201\}, \\ 16, & \text{if } \boldsymbol{r} = 1012010. \end{cases}$$

Then we have

$$T(\boldsymbol{r}, n) = \begin{cases} \left\lfloor \frac{n - n_2(\boldsymbol{r})}{6} \right\rfloor + 3 & \text{if } n \geqslant n_2(\boldsymbol{r}), \\ 2, & \text{if } |\boldsymbol{r}| + 3 \leqslant n < n_2(\boldsymbol{r}). \end{cases}$$

Our combinatorial characterisation also improves the upper bound on $T(n)$.

Theorem 3 (Upper Bound)**.** *Let R be as defined in Proposition 2, $I(i, m)$ denote the number of irreducible words in $\mathrm{Irr}_{\leqslant 3}(i, 3)$ with exactly m regions, and $U(n, i, m)$ be defined as follows:*

$$U(n, i, m) \triangleq \begin{cases} \binom{(n-i)/3+m}{m} - \binom{(n-i)/3+m-1}{m-1} + 1, & \text{if } 3 | (n-i), \\ \binom{\lfloor (n-i)/3 \rfloor + m}{m}, & \text{otherwise.} \end{cases}$$

Then

$$T(n) \leqslant \sum_{\boldsymbol{r} \in R} T(n, \boldsymbol{r}) + \sum_{i=5}^{n} \sum_{m=2}^{i} I(i, m) U(n, i, m). \tag{1}$$

In addition to the above constructions, we construct tandem-duplication codes for small lengths by searching for them exhaustively. We do so by using our confusability algorithm to construct a graph $\mathcal{G}(n, \boldsymbol{r})$, whose vertices[1] correspond to the set of all descendants of $\boldsymbol{r}$ of length n. Then $T(n, \boldsymbol{r})$ is given by the maximum size of a clique in $\mathcal{G}(n, \boldsymbol{r})$ and we use the exact algorithm MaxCliqueDyn [9] to compute $T(n, \boldsymbol{r})$ for $n \leqslant 20$. We tabulate the results $T(n)$ in Table I.

Finally, we develop certain recursive constructions.

Proposition 4. *Let $\boldsymbol{r} = r_1 r_2 \cdots r_i \in \mathrm{Irr}_{\leqslant 3}(i, 3)$. Then the following holds.*

$$T(n, \boldsymbol{r}) \geqslant \begin{cases} T(n-1, \boldsymbol{r} \setminus r_1), & \text{if } r_1 = r_3, \\ \max\{2T(n-4, \boldsymbol{r} \setminus r_1), 3T(n-8, \boldsymbol{r} \setminus r_1)\}, & \\ & \text{if } r_1 \neq r_3,\ r_1 \neq r_4, \\ \max\{2T(n-5, \boldsymbol{r} \setminus r_1 r_2), 3T(n-10, \boldsymbol{r} \setminus r_1 r_2)\}, & \\ & \text{if } r_1 \neq r_3,\ r_1 = r_4,\ r_2 \neq r_5, \\ \max\{2T(n-6, \boldsymbol{r} \setminus r_1 r_2 r_3), 3T(n-12, \boldsymbol{r} \setminus r_1 r_2 r_3)\}, & \\ & \text{if } r_1 \neq r_3,\ r_1 = r_4,\ r_2 = r_5. \end{cases}$$

Furthermore, $T(n, \boldsymbol{r}) \geqslant T(n-1, \boldsymbol{r})$ and $T(n, \boldsymbol{r}) = T(n, \boldsymbol{r}^R)$, where $\boldsymbol{z}^R$ denotes the reverse of word $\boldsymbol{z}$.

Using Proposition 4 with Proposition 2 and the values computed by MaxCliqueDyn, we derive lower bounds for $T(n)$ for $21 \leqslant n \leqslant 30$. The results are summarized in Table I. In addition to the lower bounds for the code size $T(n)$, we also compare the upper bounds in Proposition 1 and (1). Observe that (1) is tight up to lengths at most ten and the constructions in this paper improve the rates[2] for Construction 1 by as much as 6.74%.

References

[1] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh *et al.*, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, no. 6822, pp. 860–921, 2001.

[2] N. I. Mundy and A. J. Helbig, "Origin and evolution of tandem repeats in the mitochondrial DNA control region of shrikes (Lanius spp.)," *Journal of Molecular Evolution*, vol. 59, no. 2, pp. 250–257, 2004.

[3] G. R. Sutherland and R. I. Richards, "Simple tandem DNA repeats and human genetic disease." *Proceedings of the National Academy of Sciences*, vol. 92, no. 9, pp. 3636–3641, 1995.

[4] M. Arita and Y. Ohashi, "Secret signatures inside genomic DNA," *Biotechnology progress*, vol. 20, no. 5, pp. 1605–1607, 2004.

[5] D. Heider and A. Barnekow, "DNA-based watermarks using the DNA-Crypt algorithm," *BMC Bioinformatics*, vol. 8, no. 1, p. 176, 2007.

[6] M. Liss, D. Daubert, K. Brunner, K. Kliche, U. Hammes, A. Leiherer, and R. Wagner, "Embedding permanent watermarks in synthetic genes," *PloS one*, vol. 7, no. 8, p. e42465, 2012.

[7] S. Jain, F. Farnoud, M. Schwartz, and J. Bruck, "Duplication-correcting codes for data storage in the DNA of living organisms," *IEEE Transactions on Information Theory*, vol. 63, no. 8, pp. 4996–5010, 2017.

[8] Y. M. Chee, J. Chrisnata, H. M. Kiah, and T. T. Nguyen, "Deciding the confusability of words under tandem repeats," *preprint arXiv:1707.03956*, 2017.

[9] J. Konc and D. Janezic, "An improved branch and bound algorithm for the maximum clique problem," *proteins*, vol. 4, no. 5, 2007. [Online]. Available: http://insilab.org/maxclique/

[1] The number of vertices can be dramatically reduced using a finer analysis. We refer the reader to [8] for more details.

[2] The rate of a code $\mathcal{C}$ of length n is given by $(\log_3 |\mathcal{C}|)/n$.