Coding over Sets for DNA Storage

Andreas Lenz[*], **Paul H. Siegel**[†], **Antonia Wachter-Zeh**[*], and **Eitan Yaakobi**[‡]
[*]Institute for Communications Engineering, Technical University of Munich, Germany
[†]Department of Electrical and Computer Engineering, CMRR, University of California, San Diego, California
[‡]Computer Science Department, Technion – Israel Institute of Technology, Haifa, Israel
Emails: andreas.lenz@mytum.de, psiegel@ucsd.edu, antonia.wachter-zeh@tum.de, yaakobi@cs.technion.ac.il

***Abstract*—In this paper we study error-correcting codes for the storage of data in synthetic deoxyribonucleic acid (DNA). We investigate a storage model where data is stored in an unordered set of M sequences, each of length L. Errors within that model are a loss of whole sequences and point errors inside the sequences, such as insertions, deletions and substitutions. We derive Gilbert-Varshamov lower bounds and sphere packing upper bounds on achievable cardinalities of error-correcting codes within this storage model. We further propose explicit code constructions than can correct errors in such a storage system that can be encoded and decoded efficiently. Comparing the sizes of these codes to the upper bounds, we show that many of the constructions are close to optimal.**

I. Introduction

Consider a DNA-based storage system, which consists of the three following important entities: (1) a DNA synthesizer that produces the strands that encode the data to be stored in DNA; (2) a storage container with compartments that store the DNA strands; (3) a DNA sequencer that reads the strands and transfers them back to digital data. In the last few years, there has been a variety of experiments [1]–[8], which demonstrate the potential of *in vitro* DNA storage, storing up to 200MB of data [8]. On the other side, coding theoretic aspects of DNA-based data storage have, among others, been studied in [9] where a channel from a stored DNA sequence to a set of its possibly erroneous substrings is investigated. In [10] unordered multisets with errors that affect whole sequences have been discussed. Furthermore, the model proposed in this work has already been adopted in [11], [12]. Namely, codes and bounds for an arbitrary number of substitution errors in sets of DNA strands have been derived in [11]. In [12], a distance measure for the DNA storage channel has been discussed and Singleton-like and Plotkin-like code size upper bounds have been derived. In contrast to these works we propose code constructions and derive bounds for data storage in unordered sets under a loss of sequences and arbitrary point errors, such as substitutions, insertions and deletions inside the sequences. This work has been published in [13].

II. Channel Model

Storing the DNA sequences in a container removes all inherent information about their ordering, and thus the data is stored in an unordered *set* of M distinct sequences $\mathbf{x}_i \in \Sigma_2^L$

$$\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\} \subseteq \Sigma_2^L,$$

where $\mathbf{x}_i \neq \mathbf{x}_j$ for $i \neq j$. The set of all possible data sets is denoted by $\mathcal{X}_M^L$, which contains all subsets of Σ_2^L with cardinality M. For brevity we present the results in this work for binary sequences, however the extension to non-binary (and quaternary) sequences is possible as explained later. The DNA storage channel consists of the three following stages.

I. Random sequences are drawn with replacement from $\mathcal{S}$ and are sequenced, possibly with errors.
II. The sequenced strands are clustered according to their Levenshtein distance.
III. The clustered sequences are reconstructed by performing an estimate $\mathbf{x}'$ for each cluster and form the received set $\mathcal{S}'$ after removing duplicates.

We consider the combination of the above three stages as the DNA storage channel. Each sequence $\mathbf{x} \in \mathcal{S}$ is therefore either

- reconstructed without errors ($\mathbf{x} \in \mathcal{U}$),
- never drawn or its cluster is not identified ($\mathbf{x} \in \mathcal{L}$), or
- reconstructed with errors ($\mathbf{x} \in \mathcal{F}$),

where $(\mathcal{U}, \mathcal{L}, \mathcal{F})$ is a partition of $\mathcal{S}$. According to the above three cases, the three parameters $(s, t, \epsilon)_{\mathbb{E}}$ characterize the DNA storage channel, where s is the maximum number of sequences that are never drawn (or their clusters are not identified) and t denotes the maximum number of sequences, which have been reconstructed with a maximum of ϵ point errors of type $\mathbb{E}$ each. Note that a part of the errors is already corrected by the sequence reconstruction. Hence, the parameters s, t and ϵ are influenced by the number of drawn sequences. Possible error types $\mathbb{E}$ are insertions ($\mathbb{I}$), deletions ($\mathbb{D}$) and substitutions ($\mathbb{S}$). According to this channel, each possible received set is composed of error-free sequences $\mathcal{U}$ and erroneous sequences $\mathcal{F}'$, i.e., $\mathcal{S}' = \mathcal{U} \cup \mathcal{F}'$ with

$$\mathcal{F}' = \bigcup_{i=1}^{t} \{\mathbf{x}_i'\}, \mathbf{x}_i' \in B_\epsilon^{\mathbb{E}}(\mathbf{x}_{f_i}),$$

for an arbitrary partition $(\mathcal{U}, \mathcal{L}, \mathcal{F})$ of $\mathcal{S}$ with $|\mathcal{L}| \leq s$, $|\mathcal{F}| \leq t$. Here $B_\epsilon^{\mathbb{E}}(\mathbf{x})$ denotes the error ball of possible sequences after ϵ errors of type $\mathbb{E}$ around the sequence $\mathbf{x}$. $\mathcal{F} = \{\mathbf{x}_{f_1}, \ldots, \mathbf{x}_{f_t}\} \subseteq \mathcal{S}$ denotes the set of stored sequences, which are received in error and $\mathcal{F}'$ with $|\mathcal{F}'| \leq t$ is the set of all distinct erroneous received sequences $\mathbf{x}_i'$, after removing duplicates. The number of distinct received sequences $|\mathcal{S}'|$ therefore satisfies $M - t - s \leq |\mathcal{S}'| \leq M$. We define an *error-correcting code* as follows.

Definition 1. *A code $\mathcal{C} \subseteq \mathcal{X}_M^L$ is called an $(s, t, \epsilon)_{\mathbb{E}}$-**correcting code**, if it can correct a loss of s (or fewer) sequences and ϵ (or fewer) errors of type $\mathbb{E}$ in each of t (or fewer) sequences, i.e., for any possible received set $\mathcal{S}'$, it is possible to uniquely*

identify the stored set $\mathcal{S} \in \mathcal{C}$. We say $\mathcal{C} \subseteq \mathcal{X}_M^L$ is an $(s,t,\bullet)_{\mathbb{E}}$-correcting code, if the number of errors ϵ in the t erroneous sequences can be arbitrarily large. The redundancy of a code $\mathcal{C}$ is defined as $r(\mathcal{C}) = \log \binom{2^L}{M} - \log|\mathcal{C}|$.

By this definition, a *code* is a set of *codewords*, where each *codeword* is again a set of M *sequences* of length L. Table I summarizes our results with respect to upper and lower bounds on the redundancy of each case studied in the paper. For non-binary alphabets similar bounds can be obtained by employing expressions for the non-binary error-balls.

TABLE I: Lower and upper bounds on the redundancy of optimal $(s,t,\epsilon)_{\mathbb{E}}$-correcting codes. Low order terms are omitted.

Error correction	Gilbert-Varshamov bound	Sphere packing bound
$(s,t,\bullet)_{\mathbb{IDS}}$	$(s+2t)L + (s+2t)\log M$	$(s+t)L + t\log M$
$(\sigma M, \tau M, \bullet)_{\mathbb{IDS}}$	$(\sigma+2\tau)M(L-\log M)$	$(\sigma+\tau)M(L-\log M)$
$(s,t,\epsilon)_{\mathbb{S}}$	$sL + (s+2t)\log M + 2t\epsilon\log L$	$sL + t\log M + t\epsilon\log L$
$(s,t,\epsilon)_{\mathbb{D}}$	$sL + (s+t)\log M + 2t\epsilon\log(L/2)$	$sL + t\epsilon\log L$
$(0,M,\epsilon)_{\mathbb{S}}$	$2M\epsilon\log L$	$M\epsilon\log L$
$(0,M,1)_{\mathbb{ID}}$	$2M\log L$	$M\log L$

Remark 1. *Our channel model also applies to the important scenario, where the number of sequences drawn from the storage medium is significantly larger than M. Then, it can be assumed that there are enough draws per sequence, such that the reading errors are corrected by the reconstruction algorithm. Consequently, there only remain synthesizing errors, that can be modeled as in our channel model.*

III. Code Constructions

We now present two $(s,t,\bullet)_{\mathbb{IDS}}$-correcting code constructions that are suitable for correcting an arbitrary number of errors per sequence. The first construction is based on indexing each sequence to combat the loss of ordering and using a maximum distance separable (MDS) code over the sequences.

Construction 1. *For all M, L, and a positive integer δ, let $\mathcal{C}_1(M,L,\delta)$ be the code defined by*

$$\mathcal{C}_1(M,L,\delta) = \{\mathcal{S} = \{\mathbf{x}_1,\ldots,\mathbf{x}_M\} : \mathbf{x}_i = (\mathbf{I}(i),\mathbf{u}_i), \\ (\mathbf{u}_1,\ldots,\mathbf{u}_M) \in \mathsf{MDS}[M, M-\delta]\}.$$

where $\mathbf{I}(i) \in \Sigma_2^{\lceil \log M \rceil}$ is a binary representation of the index i and $\mathsf{MDS}[M, M-\delta]$ is a MDS code of redundancy δ.

It can be verified, that $\mathcal{C}_1$ is $(s,t,\bullet)_{\mathbb{IDS}}$-correcting for all $s+2t \leq \delta$, $(s,t,\bullet)_{\mathbb{I}}$-correcting or $(s,t,\bullet)_{\mathbb{D}}$-correcting for all $s+t \leq \delta$. The redundancy of Construction 1 is given by

$$r(\mathcal{C}_1(M,L,\delta)) = M\log \mathrm{e} + \delta(L - \lceil \log M \rceil) + o(1).$$

It is possible to obtain a construction with lower redundancy by using a truncated representation of the index. Thereby, some sequences share the same index and the sets of their data fields $\mathbf{u}_i$ can be combined to one symbol of a MDS code, similar to Construction 1. The second construction uses a binary representation $\mathbf{v}(\mathcal{S}) \in \Sigma_2^{2^L}$ of a set $\mathcal{S}$, where each non-zero entry in $\mathbf{v}(\mathcal{S})$ indicates that a specific sequence is contained in the set $\mathcal{S}$. Using this representation, a loss of a sequence $\mathbf{x} \in \mathcal{S}$ corresponds to a $1 \to 0$ error in $\mathbf{v}(\mathcal{S})$ at the position corresponding to $\mathbf{x}$. Substitution errors inside a sequence $\mathbf{x} \in \mathcal{S}$ translate to an error in the Johnson graph in $\mathbf{v}(\mathcal{S})$, i.e., a $1 \to 0$ error at the position of the original sequence $\mathbf{x}$, and a $0 \to 1$ error at the position of its outcome $\mathbf{x}'$. This principle provides the following construction.

Construction 2. *For all M, L and positive integers s, t, let $\mathcal{C}_M^L(s,t) \subseteq \Sigma_2^{2^L}$ be a code that consists of codewords with constant Hamming weight M of length 2^L, which corrects s asymmetric $1 \to 0$ errors and t errors in the Johnson graph. We then define the following code*

$$\mathcal{C}_2(M,L,s,t) = \{\mathcal{S} \in \mathcal{X}_M^L : \mathbf{v}(\mathcal{S}) \in \mathcal{C}_M^L(s,t)\}.$$

Due to the translation of a loss of sequences and errors in sequences to asymmetric errors and errors in the Johnson graph, the code $\mathcal{C}_2(M,L,s,t)$ is $(s,t,\bullet)_{\mathbb{IDS}}$-correcting. Using a coset of a binary alternant code with sufficiently many words of weight M for the code $\mathcal{C}_M^L(s,t)$, it can be shown that there exists a code $\mathcal{C}_2(M,L,s,t)$ with redundancy at most

$$r(\mathcal{C}_2(M,L,s,t)) \leq (s+2t)L.$$

The redundancy of Construction 2 is smaller than that of Construction 1, especially for the considered case $M \gg L$. However, for Construction 1 there exist efficient encoders and decoders while this is unclear for Construction 2, since the code length of the constant-weight code is exponential in L. Both constructions can be extended to non-binary alphabets by defining the MDS code in Construction 1 over a different field and adapting the length of $\mathbf{v}(\mathcal{S})$ in Construction 2.

References

[1] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, no. 6102, pp. 1628–1628, Sep. 2012.

[2] N. Goldman *et al.*, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, no. 7435, pp. 77–80, Jan. 2013.

[3] R. N. Grass *et al.*, "Robust chemical preservation of digital information on DNA in silica with error-correcting codes," *Angewandte Chemie Int. Edition*, no. 8, pp. 2552–2555, Feb. 2015.

[4] S. M. H. T. Yazdi *et al.*, "A rewritable, random-access DNA-based storage system," *Nature Scientific Reports*, no. 14138, Aug. 2015.

[5] J. Bornholt *et al.*, "A DNA-based archival storage system," in *Proc. 21st Int. Conf. Architectural Support for Programming Languages and Operating Systems*, Atlanta, Apr. 2016, pp. 637–649.

[6] M. Blawat *et al.*, "Forward error correction for DNA data storage," in *Int. Conf. Computational Science*, San Diego, Jun. 2016, pp. 1011–1022.

[7] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, no. 6328, pp. 950–954, Mar. 2017.

[8] L. Organick *et al.*, "Random access in large-scale DNA data storage," *Nature*, pp. 242–248, Mar. 2018.

[9] H. M. Kiah, G. J. Puleo, and O. Milenkovic, "Codes for DNA sequence profiles," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3125–3146, Jun. 2016.

[10] M. Kovačević and V. Y. F. Tan, "Codes in the space of multisets – coding for permutation channels with impairments," *IEEE Trans. Inf. Theory*, no. 7, pp. 5156–5169, Jul. 2018.

[11] J. Sima, N. Raviv, and J. Bruck, "On coding over sliced information," 2018. [Online]. Available: http://arxiv.org/abs/1809.02716

[12] W. Song and K. Cai, "Sequence-subset distance and coding for error control in DNA-based data storage," 2018. [Online]. Available: http://arxiv.org/abs/1809.05821

[13] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "Coding over sets for DNA storage," 2018, submitted to *IEEE Trans. Inform. Theory*. [Online]. Available: https://arxiv.org/abs/1812.02936