Encoding Schemes for DNA Data Storage Based on the Assembly of Shortmers

Kaya Wernhart, Fabian Schroeder, Mathias Orlando, Julian Scherer, Ivan Barisic

2nd November 2024

1 Introduction

DNA is the information storage medium of nature. It is highly dense ($4.6 \times 10^8 GB/mm^3$)[12] and extremely stable (1 cut/century/10^5nt)[10] and, thus, has the potential to solve mankinds increasingly pressing (cold) data storage needs efficiently and sustainably [5]. The idea of using DNA as a storage medium has been discussed as early as the mid-1960s [24], however, only in the early 2010s has the storage of substantial amounts of data in DNA molecules been demonstrated [9, 17]. These advancements, leveraging next-generation technologies for both DNA synthesis and sequencing, propelled DNA data storage into the spotlight, showcasing its high capacity, stability, and potential to revolutionize data storage paradigms. The costs have decreased from 0.1$ per nucleotide in the 1980s to less than $1 \cdot 10^{-4}$$ per nucleotide in today ($1 \cdot 10^8$$ per TB at 1 bit / bp) [25]. The time required to sequence a TB has decreased from several years to a few days [18]. Despite these advances of about 3 orders of magnitude in both costs and time, the technology is still far from being a valid business case. In contrast, tape storage, a widely used conventional method, costs about 16$ per TB, with prices decreasing by 10% annually [5, 14]. Thus, to become a viable alternative to conventional storage technologies, DNA data storage must decrease costs by several orders of magnitude, in particular the synthesis as it is by far the most costly step in the data storage workflow.

A potential alternative to DNA synthesis on the nucleotide level is the assembly of DNA shortmers, which involves the precise joining of shorter oligonucleotide sequences to create longer, contiguous DNA strands. These strands often reach gene-length scales (1-15kbp), which is why the construction process is often referred to as "gene synthesis". This synthesis of gene-length DNA is crucial for constructing functional genetic elements for research and biotechnological applications, and, thus, techniques such as Gibson assembly or Golden Gate assembly have been established. The mechanistic details of these methods have been reviewed extensively elsewhere [4, 6, 11].

Assembly-based DNA data storage storage offers the potential to reduce the costs of DNA synthesis dramatically, given that a finite set of DNA fragments can be produced, amplified, and assembled in a cost-effective manner. It further does not involve any toxic chemicals and can, thus, greatly reduce the environmental footprint compared to chemical de-novo synthesis. In this paper, we will systematically analyse the suitablity of DNA assembly for DNA data storage purposes. More precisely, we will explore different algorithms and coding schemes for this undertaking and evaluate their performance in terms of (i) thermodynamic properties, (ii) time complexity, (iii) cost, and (iv) environmental impact. Particular attention will be paid to the parallelisation / scalability of the encoding schemes as these affect time and cost.

2 Assembly-based DNA Encoding Schemes

Essentially, all assembly methods rely on the hybridization of complementary single stand (ss) DNA stretches and the subsequent ligation of the backbone. Thus, in this section, we will abstract from the biochemical details and focus on the algorithmic aspects of DNA assembly for DNA data storage. For this purpose, let us define an *assembly-based DNA encoding scheme* as a pair $(\mathscr{A}, \psi)$, where $\mathscr{A}$ denotes a *library of oligonucleotides*, and ψ an *encoding* *function* that maps a bit sequence to a sequence of elements of $\mathscr{A}$. More formally, a library is a set of short oligonucleotides (shortmers) $\mathscr{A} \subseteq \Omega_n := \{A,C,G,T\}^n$ [1], which can be assembled in an arbitrary order to encode information using only elements of $\mathscr{A}$. For the *half overlapping ligation scheme* (HOLS) depicted in Figure 2, this means that every pair a, b of elements of $\mathscr{A}$ can be ligated by an oligo that is the reverse complement of the second half of a and the first half of b. $\mathscr{A}$ is a library for the HOLS $:\iff \forall a,b \in \mathscr{A} : (b_{n/2-1}, b_{n/2-2}, \ldots, b_1, a_n, a_{n-1}, \ldots, a_{n/2+1}) \in \mathscr{A}$. $\psi : \{0,1\}^m \mapsto \mathscr{A}^n$ is a bijection, ensuring that any bit sequence can be mapped to a unique sequence of oligos and vice versa.

2.1 Library size

Most data storage schemes rely on the encoding of information in the sequence of nucleotides, since they are based on synthesis on the nucleotide level. In theory, this is also possible based on the assembly of oligos from the *complete library* of n-mers $\mathscr{A}_n = \{A,C,G,T\}^n$. Since such a library is exhaustive, every possible nucleotide sequence can be obtained by assembly and, thus, the theoretical maximum of 2 bits per nucleotide can be reached. However, such a library must contain 4^n different oligos, which becomes infeasible for even moderate oligo lengths. Equation 1 states a simple relationship between the size a library $|\mathscr{A}|$ and the upper bound of the information density per nucleotide - obtained only when all oligos encode for information.

$$d = \frac{\log_2(|\mathscr{A}|)}{n} \tag{1}$$

The nominator is simply the *Shannon Entropy* of a uniformly distributed random variable on the library $\mathscr{A}$, while the denominator is the length of the oligos n. Due to the logarithmic relationship, it is possible to reduce the number of oligos in the library (compared with the complete library) by several order of magnitude while still achieving a reasonable information density. With e.g. 256 oligos of length 30, a density of 0.27 bits per nucleotide can be achieved, which equals a reduction by a factor of 7.5 (less than 1 order of magnitude) whereas the reduction in the number of oligos is 15 orders of magnitude ($4.5 \cdot 10^{15}$).

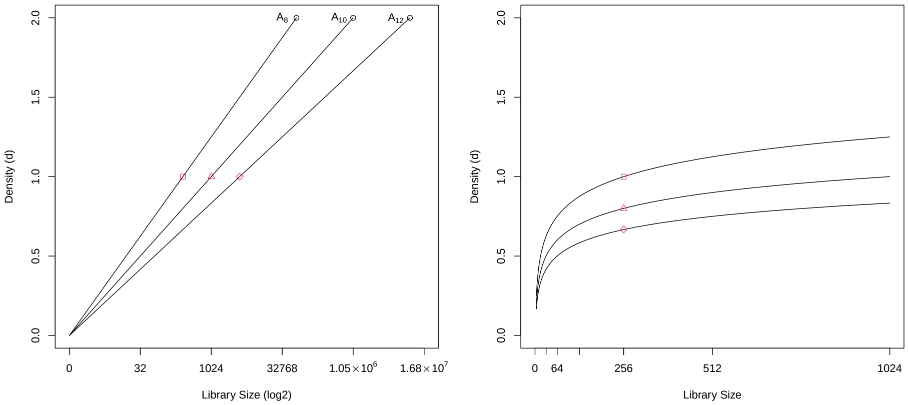


Figure 1: An illustration of the upper bound of the information density depending on the library size on both a linear and a log scale. The three designated points correspond to libraries of size 256 of oligos of length 8, 10, and 12, respectively.

In most DNA data storage schemes, the smallest unit of information is a nucleotide. The use of non-complete libraries, however, requires alternative encoding strategies.

We propose a family of encoding schemes where the smallest unit of information is a *motif*. This is similar to a codon in genetics, where a tripple of nucleotides encodes for one protein. Consider a library, in which every

[1] implicit assumption of same length nucleotides

oligo can be represented as a concatenation of two motifs - a left and a right one.[2] There are, thus, two equivalent ways to represent the elements of an alphabet: (i) by their nucleotide sequence and (ii) by their symbol tuple $\forall a \in \mathscr{A} : a \sim (s_i, s_j) : s_i \in S_l, s_j \in S_r \in \mathrm{S}$, where S_l and S_r are the set of left and right motifs, respectively. A codeword is assembled by a chain of "half overlapping oligos" as illustrated in Figure 2.

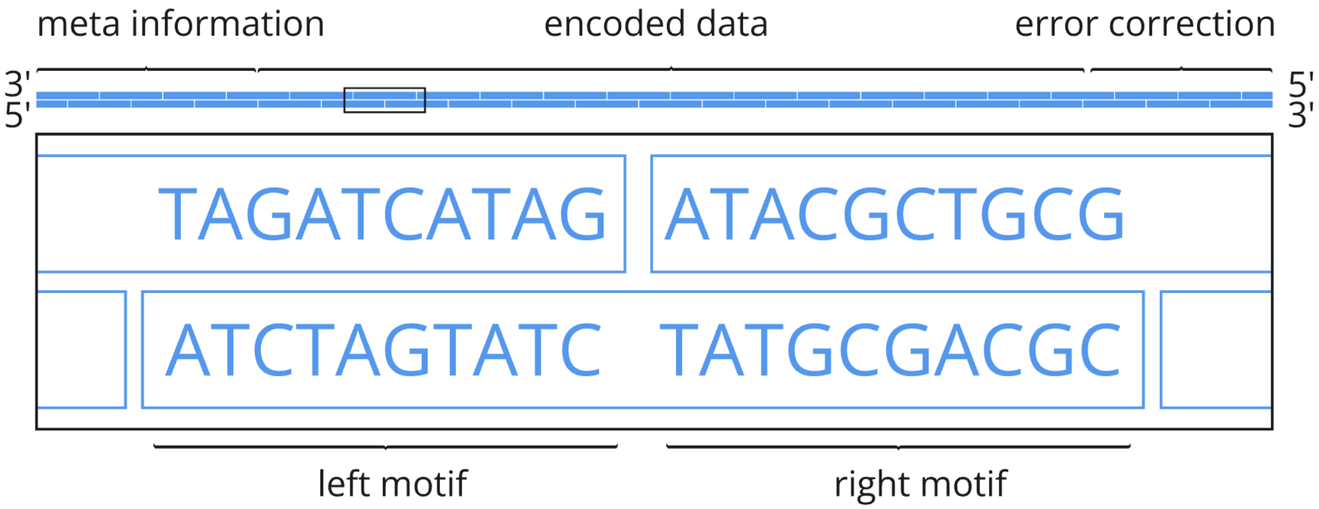


Figure 2: An illustration of a simple ligation scheme of 20mers, where every oligo consists of a left and a right motif. A codeword is assembled by a chain of such two motif oligos, where the first section consists of meta information about the data block and the final section contains error correction information.

For such a library to be useful for data encoding, it must be *generative* - meaning that an arbitrary sequence of oligos can be obtained by binding oligos from the library. This can be achieved by a simple construction: (i) define arbitrary sets of left and right motifs, L and R, of length n respectively, (ii) unite these sets with their respective sets of reverse complements L^c and R^c, and (iii) concatenate the elements of the cartesian product of these sets. This simple construction is described in the supplementary material (6.1).

There are many possible ways to define encoding functions for such a library, however, the simplest way is to index the library $i = 1, \ldots, 2^n$ and to map a bit sequence to the oligo with the index number given by the decimal representation of the bit sequence. We will use this encoding scheme based on a library of $2^8 = 256$ oligos.

$$\psi : \{0,1\}^{m_1} \mapsto \mathscr{A}^{m_2} : \psi(b_0, \ldots, b_{m_1}) = o_{\sum_i^{m_1} b_i 2^i} \tag{2}$$

2.2 Time complexity of the assembly process

Such an encoding scheme is optimal from the perspective of information density for libraries of a given size, since all oligo encode information. However, data density is not the only relevant factor for the performance of a DNA data storage system. The time complexity of the assembly process is another crucial factor. For the hybridisation of a set of oligos to be unambiguous, the oligos in any reaction chamber must contain every motif not more than once. There are different pooling strategies to achieve this. The simplest one is to ligate the oligos in a serial manner, ligating only as many oligos per pool as are possible without motif-repetition. However, this would require the

If we denote the time required for a single hybridisation and ligation step as t, we can express the required number of pooling steps for a given data package as a function of t. Since the sequence of oligos is stochastic, the expected value of the number of oligos per pool is XXX and, thus, the number of bits per pool is YYY. The distribution of the number of oligos per pool as well as the expected value is derived in the supplementary material (6.2).

[2] We could consider oligos that consist of more than two motifs. However, for a given library size, this does not make any sense.

A pooling strategy that reduces the time complexity of the assembly process is the hierarchical pooling. On the first level, oligos are ligated in pools of varying size such that no motif repetition is possible. On the second level, the oligos from the first level are ligated in pools of the same size. Again, the number of double stranded oligo in each pool is limited by that fact that two "sticky ends" must not have the same motif in order to avoid unambiguous binding. The time complexity of this pooling strategy is ZZZ, where is the number of motifs in the right and left motif set, respectively???. The exact number of pooling steps and the expected value of the number of oligos per pool is derived in the supplementary material (**??**).

There are, however, possible approaches to increase the bits per time unit by trading-off data density. One such possibility is to introduce positional oligos, which sole purpose is the ensure the correct position of neighboring oligos in the oligo. Obviously, these oligos will not encode any information and, thus, the upper bound for a given library size cannot be obtained. In other words, there is a trade-off between time units and data density. The ligation scheme illustrated in Figure 3 reduces the density by $1/3$, but enables that entire codewords (with a bitsize of ZZZ) can be assembled in only two steps. The bitrate per time unit is, thus, XXX. Further, the assembly strategy is independent of the data encoded, which reduces the complexity of the engineering task.

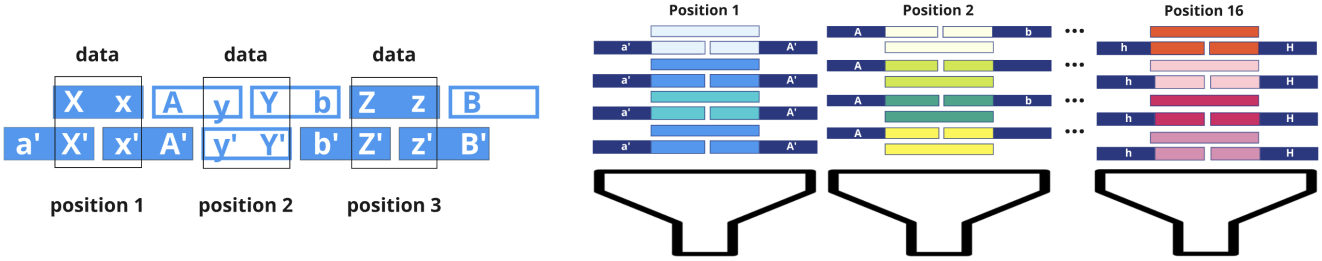


Figure 3: Schematic representation of an assembly strategy built on positional oligos and message oligos. The message oligos encode the actual information, while the positional oligos ensure the correct position of the message oligos.

The advantage, however, is that the parallel pooling also enables combinatorial (or binomial) encoding - an approach that can increase the data density by encoding data not in the sequence of nucleotides / oligos but in mixtures of such entities [27]. If at a certain position there are n possible oligos, then normal encoding can encode $\log 2(n)$ bits at that position, e.g. 8 bits for libraries with $n = 256$ oligos. If, however, our information is encoded by a selection of k out of n elements at that position, then that position can encode $\log_2(\binom{n}{k})$ bits of data, where $\binom{n}{k}$ denotes the binomical coefficient. This relationship is depicted in the left part of Figure 12 for $k = 5$ in the supplementary material (6.4).

Binomial encoding can bring significant improvements in data density, however, it only makes sense, if different variants of an oligo can be assembled "at no additional cost" in terms of time. More specifically, this would mean, that different variants of a code word could be assembled in parallel and would not require more time or more reaction pools than creating a single code word. Such a assembly / encoding strategy in depicted in Figure 13. The number of bits that can effictively be encoded is, thus, $\lfloor \log_2((\binom{(m-1)^2}{k}))\rfloor$, where m is the number of (left and right) motifs in the library. Its derivation and additional results can be found in the supplementary material (6.4).

An altogether different assembly strategy is to completely decouple the positional and message oligos (drawing them from different sets). This allows to have an arbitrary number of message oligos but requires the use of Polymerase to fill double stranded DNA..... This approach was explored by Catalog, a company that has developed end-to-end DNA data storage solutions.

2.3 Thermodynamic optimization of oligo sequences

Until now, we abstracted from the biochemical details of DNA assembly and represented oligos as strings of nucleotides and only considered the complementarity of A, T and C, G. For algorithmic purposes, designing good assembly and pooling schemes, this level of detail is sufficient. For the purpose of identifying optimal sequences

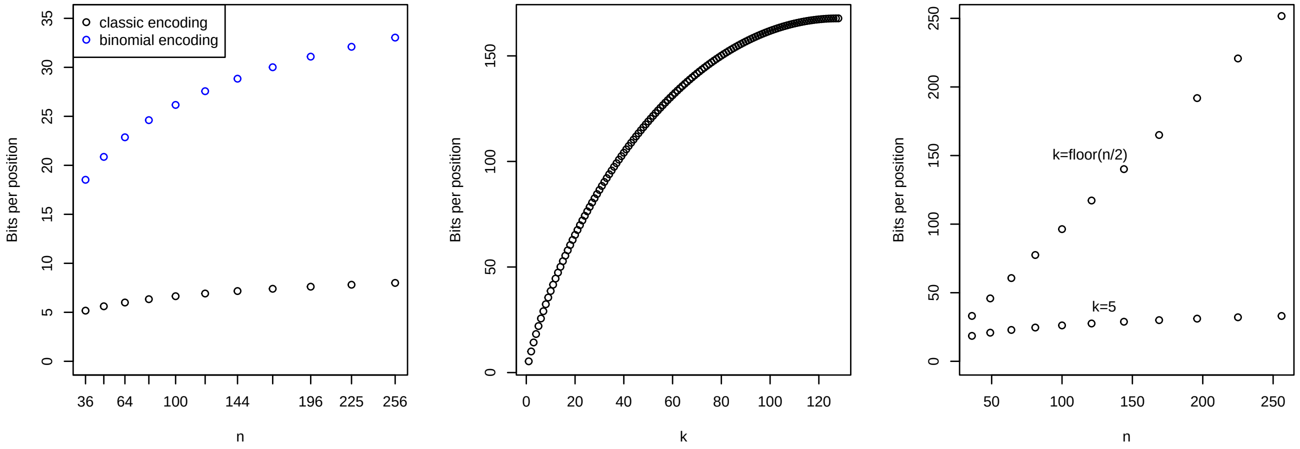


Figure 4: Analysis of the encoding bitrate as a function of n and k.

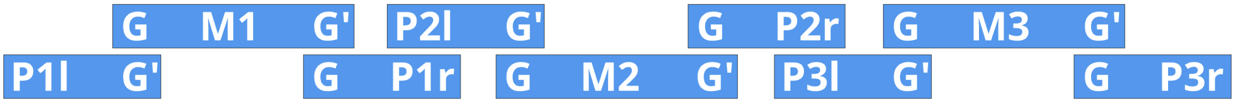


Figure 5: Schematic representation of an assembly strategy built on disjoint sets of positional and message oligos with different lengths.

that exhibit good binding behaviour, we need to consider the thermodynamic properties of DNA in greater detail. To achieve a highly controlled DNA assembly under standardised conditions, the oligos of a library need to exhibit similar thermodynamic behaviour. The characteristics of hybridisation between oligos are not only determined by the GC content, but also by neighborhood- and stacking effects [1 ???]. Hence, various computational models have been developed to predict melting temepratures of oligos, with nearest-neighbor thermodynamic model (NNM) being one of the most widely used [2 ???]. For this purpose, let us formulate the following hypothesis: For any given oligo length n and any library size m, it is possible to identify a generative library that exhibits (i) a clear separation in terms of free energy between intended bindings and unintended binding and (ii) avoids homopolymers and CG imbalance. Criteria (i) is essential to ensure that the oligos can be assembled unambiguously. Criteria (ii) is crucial for the effectiveness of the sequencing process, as homopolymers and imbalanced CG-content can cause higher error rates in sequencing techniques [26]. This problem can be restated as an optimization problem:

$$\max_{A\subseteq\Omega_n}\left|\max_{x\in I(A)} G(x) - \min_{y\in U(A)} G(y)\right| \tag{3}$$

subject to the following constraints: (i) $\mathscr{A}$ is a library, (ii) the elements of $\mathscr{A}$ exhibit balanced CG content, and (iii) the elements of $\mathscr{A}$ do not contain homopolymers. $G(x)$ denotes the Gibbs free energy of the conformation x and $I(A)$ and $U(A)$ are the sets of intended and unintended conformations of pairs of oligos, respectively. To validate this hypothesis in an in silico process, we will attempt to generate a valid library with these characteristics in the following iterative process described in pseudocode:

Algorithm 1: for library generation

```
while $crit < limit$ do
    $L \leftarrow \emptyset$;
    $R \leftarrow \emptyset$;
    while $|L| < \sqrt{size\ of\ library}$ and $|R| < \sqrt{size\ of\ library}$ do
        Randomly select oligos $l$ and $r$ with $n/2$ nucleotides;
        if $l$, $l^c$, $r$, and $r^c$ exhibit $\sim 50\%$ CG content and differ in at least $\frac{n}{4}$ nucleotides from all
          polynucleotides in $L$ and $R$ then
            $L \leftarrow L \cup \{l, l^c\}$;
            $R \leftarrow R \cup \{r, r^c\}$;
        end
    end
    Generate the library $\mathscr{A}(L^*, R^*)$;
    Calculate the Gibbs free energy of binding for all possible pairings of oligos in $\mathscr{A}(L^*, R^*)$;
    crit $= |\min\{\Delta G(i,j) : i,j \in \mathscr{A}(L^*, R^*)\text{ intended}\}| - |\max\{\Delta G(i,j) : i,j \in \mathscr{A}(L^*, R^*)\text{ unintended}\}|$;
end
```

As the thermodynamic model for DNA, we used the nearest-neighbour model, an improved version of the SantaLucia model [21]. As the name suggests, the model predicts the Gibbs free energy of bound nucleotides by summing up the energies of bindings in relation to the three nearest nucleotides (nearest neighbors) of a given DNA strand. It was calibrated by a large melting curve array experiment and can find the most likely secondary structure of the DNA strand given.

Figure 6 illustrates the schematic results of this in silico experiment consisting of two histograms where the Gibbs free energy of binding is plotted for all possible pairings of oligos in the library. The blue distribution represents the intended bindings (G_I), while the orange distribution represents the unintended bindings (G_U). More details can be found in (6.5).

2.4 Oligo length optimization

This experiment was repeated (25 libraries) for different oligo lengths with a constant library size in order to assess the impact of oligo length (and, thus, the size of the sequences space) on the possibility to obtain good libraries. As the oligos become longer, the number of hybridisations for intended bindings increases. At a length of 36, 38 and 40 base pair oligos the criterium of a positive gap is met, providing initial evidence that the oligo needs to be longer thanxs originally anticipated.

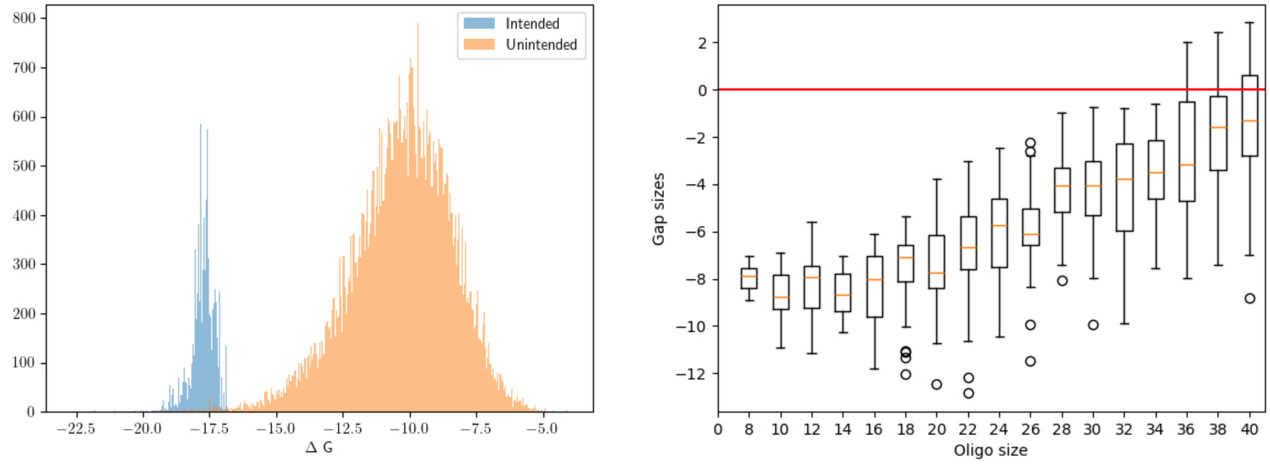


Figure 6: (a) Illustration of the results of the in silico experiment described in Section 2.3. Every bar represents how many pairs have a certain Gibbs free energy of binding, where blue represents the intended bindings and orange the unintended bindings. At the given oligo length of 36, there is a clear separation between the two distributions. (b) Boxplots of the gap sizes between ΔG_I and ΔG_U for varying oligo lengths. As the oligo length increases, the gap finally exceeds 0 at lengths of 36.

3 Experimental Validation

Despite the high accuracy of state-of-the-art computational models, experimental validation remains the gold standard for determining the thermal characteristics of oligos in different ionic conditions as well as their ligation behaviour. In order to validate the libraries experimentally, we ran two different experiments: (i) melting temperature (Tm) measurements using a qPCR machine and (ii) ligation experiments using agarose gel electrophoresis and a fragment analyser for evaluation. Furthermore, we implemented a digital twin of the pipeline to validate the end-to-end process in an in silico environment.

3.1 Melting curve experiments

To validate the predicted melting behavior (separation of intended and unintended bindings) of our designed oligo library and to detect potential problems that could complicate subsequent DNA assembly reactions, we performed melting temperature (Tm) measurements on a qPCR machine using oligos with complementary motives and an intercalating, fluorescent dye for detection. We assesses the melting temperatures of each left- and right- complementary pairs and compare the results to predicted values of IDT's OligoAnalyzer™ Tool.

Figure 7 depicts the melting curves of all 16 RCPs. Each melting curves represents the melting behavior of two different oligos with reverse complementary motives of length 10 nt. Whereas L-RCP-7 displays a "classical" trend of a melting curve with a peak at 35°C, other curves like R-RCP-6 show a plateau between 25-50°C where no clear melting peak can be obtained. This can be caused by many factors like secondary structures, influence of flanking, non-hybridizing motives or experimental shortcoming. Preliminary results indicate that Tm-predictions are not always confirmed by experimental results. Experimental replication and different designs will further advance our understanding the melting behavior of our oligo library. Material and methods for this experiment can be found in the supplementary material (**??**).

Table 1 compares the experimentally determined Tms with predicted Tms by IDT's OligoAnalyzer™ Tool with the given paramteres. For some melting curves, the Light Cycler 480® software was not able to call Tms, as indicated by "-". Notably, the majority of measured Tms deviate away from the predicted Tms more than one standard deviation. This can be caused by an imperfect experimental setup or by limitations of the predictive

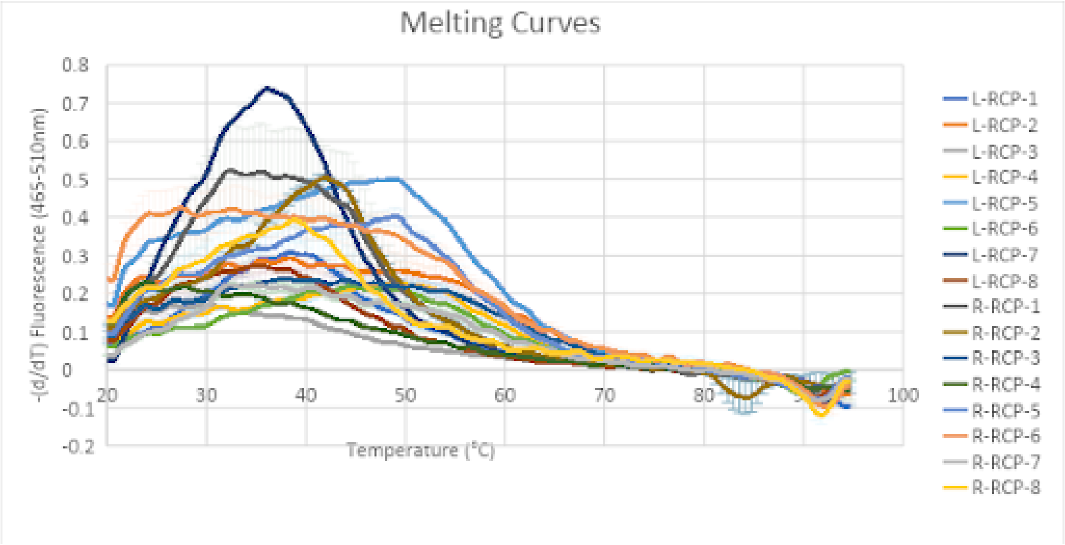


Figure 7

models. Generally speaking, repition of the described ecperiments would improve the validity of the results and different experimental design could further illumate properties of the given library. For example, one RCP could be fixed and tested with many different oligo combinations to assess the effect of the flanking, non-hybridising 10-nt sequences. It must be noted that the hybridisation of a 10 base pair stretch results is not optimal for detection, as a usual qPCR measurement detects dsDNA of much greater length (hundreds of base pairs), leading to higher signal intesities.

RCP	**Measured**	**Predicted**	**Δ measured-predicted**	
	μ(Tm) (°C)	**σ(Tm) (°C)**	**Tm (°C)**	**(°C)**
L-RCP-1	35.77	2.32	38.1	-2.33
L-RCP-2	50.44	4.24	38.5	11.94
L-RCP-3	-	-	38.1	-
L-RCP-4	51.265	4.82	38.7	12.565
L-RCP-5	50.62	1.15	39.2	11.42
L-RCP-6	43.975	0.02	39.3	4.675
L-RCP-7	34.965	0.16	38.6	-3.635
L-RCP-8	32.985	4.29	38.1	-5.115
R-RCP-1	33.63	1.64	38.7	-5.07
R-RCP-2	41.995	1.21	38.7	3.295
R-RCP-3	47.595	0.76	38.8	8.795
R-RCP-4	28.76	-	38.1	-9.34
R-RCP-5	48.245	1.21	39.2	9.045
R-RCP-6	-	-	37.8	-
R-RCP-7	35.175	0.09	38.8	-3.625
R-RCP-8	35.25	3.15	39.3	-4.05

Table 1: Comparison of measured and predicted melting temperatures (Tm) for various RCPs.

3.2 Ligation Experiments

In the second experiment, we binarized the first verse of Queen's "Bohemian Rhapsody" (XXX kB) and attempted to ligate a set of oligos from a 256-oligonucleotide library. Based on our encoding scheme, this resulted in 9,600 base pairs DNA sequence through 282 pooling steps, using a total of 960 oligonucleotides. A python function not only implemented the described encoding scheme but also generated a detailed pipetting scheme for each reaction.

The ligation products were then analysed by means of agarose gel electrophoresis (see Figure) and a fragment analyser (see Figure). Details of the experiment can be found in the supplementary material (6.6).

3.3 Digital twin for DNA data storage

To assess the performance of the encoding scheme in an application scenario, we created a digital twin of the end-to-end DNA data storage process based on our library discussed in chapter 2. This software programm allows to (i) binarize files and add error correction, (ii) encode the information in DNA oligonucleotides, simulate the process of (iii) synthesising, (iv) storing, and (v) sequencing the DNA. Further, it allows to (vi) correct errors that were generated in steps (iii-v), (vii) decode the data back into binary data, and finally (vii) restore the original files. Similar simulation pipelines have been developed in the past [2, 28, 7, 16], however in a manner that did not enable the use of our library and encoding schemes. Our comprehensive digital twin - depicted in Figure 9 - allows the user to set multiple parameters, such as the type of encoding library, error correction algorithms, storage characteristics, sequencing technology etc. In order to be easily extendible, the digital twin was written in completely modular way using the Python language and can be found under (https://github.com/EkoRefugium/mi_dna_disc).

Binarization The very first step of the pipeline is to take given input files, compress them and convert them into a binary string. Subsequently the data is split up into codewords consisting of (i) meta information and the beginning, (ii) the encoded data, and (iii) the error correction at the end. The amount of redundancy can be chosen by the user. Common error correction methods like Reed-Solomon and fountain codes were implemented and their parameters can be finetuned by the user.

Encoding Once the binary representation of the codewords for a specific encoding scheme is established, they are encoded into oligos. The user can specify (i) a library, (ii) an encoding scheme (simply, binomial, ...) and (iii) the pooling strategy. The order of oligos as well as the pooling sequence are then saved and passed on to the next step of the pipeline.

Assembly In the assembly step, the oligonucleotides for all the codewords are mixed in the order specified by the pooling algorithm. Our pooling method has two different modes, one is a theoretical mode that returns one of each code word in the perfectly hybridized form while the default mode attempts to simulate random hybridisation in a reaction chamber. The result of this simulation includes oligos of different lengths and potentially incorrectly hybridised oligos.

Storage During this step, the long-term storage of the oligonucleotides is simulated. The user can choose the storage period (in year) as well as the conditions under which the oligos are stored (Permafrost, room temperature, ideal capsulated conditions, etc.) determining the rate of strand breaks and, thus, the half-life of the data [10, 1].

Sequencing In this step, we simulate the process of reading the data by sequencing. Currently, two different sequencing technologies are implemented in the digital twin: (i) Illumina sequencing, and (ii) Oxford Nanopore sequencing. Illumina errors are not random but are strand specific and the error rates are higher at the end of a read. Substitution error rates are about 0.0015 - 0.0004 errors per base. Insertions and deletions are significantly less likely, they are on the order of 10^-6. Nanopore sequencing errors are characterized by a higher error rate compared to Illumina sequencing, with a mix of substitution, insertion, and deletion errors. Errors are more uniformly distributed across the read length compared to Illumina sequencing.

Correction After sequencing, the obtained sequences are filtered and aligned to match the expected length and structure. To accurately map the sequenced oligonucleotides to the elements of the reference library, we employ a variant of the Levenshtein distance, a metric for measuring the difference between two sequences. The Levenshtein distance quantifies the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one sequence into another. By calculating the Levenshtein distance between each sequenced oligo and every oligo in the reference library, we can identify the closest matches. This approach is particularly effective in handling sequencing errors, as it allows for a robust comparison that accounts for minor discrepancies. Once the closest matches are identified, the sequenced oligos are mapped to their corresponding library elements, facilitating subsequent error correction and data reconstruction steps.

Decoding The decoding process involves interpreting the sequenced oligonucleotides to reconstruct the original data. The sequences are then split according to the encoding scheme used, isolating the information-carrying segments of each codeword. For binomial encoding, this amounts to identifying all codewords belonging to one data block and inferring the selection of motifs at each position.

Restoration of original files In the final step, the binary error correction algorithms are applied, the data is decompressed, and the binary string is converted back into the original file format.

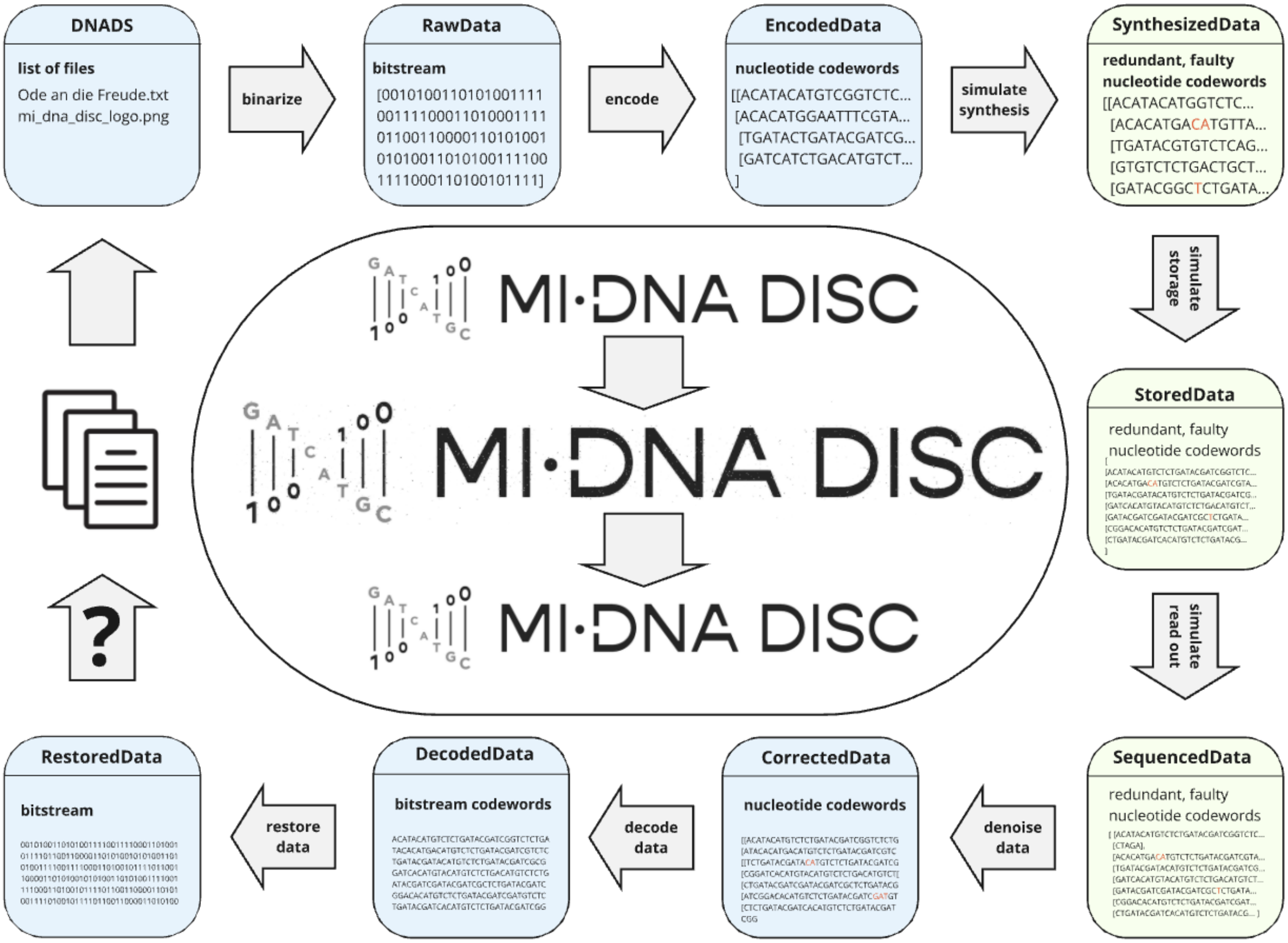

DNADS
list of files
Ode an die Freude.txt
mi_dna_disc_logo.png
binarize
RawData
bitstream
[0010100110101001111
0011110001101000111
0110011000011010100
0101001101010011110
1111000110100101111]
encode
EncodedData
nucleotide codewords
[[ACATACATGTCGGTCTC...
[ACACATGGAATTTCGTA...
[TGATACTGATACGATCG...
[GATCATCTGACATGTCT...
]
simulate synthesis
SynthesizedData
redundant, faulty nucleotide codewords
[[ACATACATGGTCTC...
[ACACATGACATGTTA...
[TGATACGTGTCTCAG...
[GTGTCTCTGACTGCT...
[GATACGGCTCTGATA...
simulate storage
StoredData
redundant, faulty nucleotide codewords
[
[ACATACATGTCTCTGATACGATCGGTCTC...
[ACACATGACATGTCTCTGATACGATCGTA...
[TGATACGATACATGTCTCTGATACGATCG...
[GATCACATGTACATGTCTCTGACATGTCT...
[GATACGATCGATACGATCGCTCTGATA...
[CGGACACATGTCTCTGATACGATCGAT...
[CTGATACGATCACATGTCTCTGATACG...
]
simulate read out
SequencedData
redundant, faulty nucleotide codewords
[[ACATACATGTCTCTGATACGATCGGTCTC...
[CTAGA],
[ACACATGACATGTCTCTGATACGATCGTA...
[TGATACGATACATGTCTCTGATACGATCG...
[GATCACATGTACATGTCTCTGACATGTCT...
[GATACGATCGATACGATCGCTCTGATA...
[CGGACACATGTCTCTGATACGATCGAT...
[CTGATACGATCACATGTCTCTGATACG...]
denoise data
CorrectedData
nucleotide codewords
[[ACATACATGTCTCTGATACGATCGGTCTCTG
[ATACACATGACATGTCTCTGATACGATCGTC
[[TCTGATACGATACATGTCTCTGATACGATCG
[CGGATCACATGTACATGTCTCTGACATGTCT[
[CTGATACGATCGATACGATCGCTCTGATACG
[ATCGGACACATGTCTCTGATACGATCGATGT
[CTCTGATACGATCACATGTCTCTGATACGAT
CGG
decode data
DecodedData
bitstream codewords
ACATACATGTCTCTGATACGATCGGTCTCTGA
TACACATGACATGTCTCTGATACGATCGTCTC
TGATACGATACATGTCTCTGATACGATCGCG
GATCACATGTACATGTCTCTGACATGTCTCTG
ATACGATCGATACGATCGCTCTGATACGATC
GGACACATGTCTCTGATACGATCGATGTCTC
TGATACGATCACATGTCTCTGATACGATCGG
restore data
RestoredData
bitstream
001010011010100111100111100011010010
011110110011000011010100101010011010
010011110011110001101001011110110010
100001101010010101001101010011110010
111000110100101111011001100001101010
001111010010111101100110000110101000
MI·DNA DISC
MI·DNA DISC
MI·DNA DISC
?

Figure 9: Schematic representation the digital twin for DNA data storage.

4 Discussion

5 Conclusion

Assembly-based DNA encoding schemes are a promising approach for DNA data storage.

There are severall sources for cost reduction for both synthesis and sequencing.

The raw material for such encoding schemes could potentially be produced in in vivo bioreactors, which have the potential to dramatically reduce the costs.

Data encoding in sequences of shorter oligos, can be sequenced at much higher error levels than in information encoded on the nucleotide granularity. This offers potentially price and time advantages for the sequencing process.

References

[1] Morten E Allentoft, Matthew Collins, David Harker, James Haile, Charlotte L Oskam, Marie L Hale, Paula F Campos, Jose A Samaniego, M Thomas P Gilbert, Eske Willerslev, et al. The half-life of dna in bone: measuring decay kinetics in 158 dated fossils. *Proceedings of the Royal Society B: Biological Sciences*, 279(1748):4724–4733, 2012.

[2] Jamie J Alnasir, Thomas Heinis, and Louis Carteron. Dna storage error simulator: A tool for simulating errors in synthesis, storage, pcr and sequencing. *arXiv preprint arXiv:2205.14437*, 2022.

[3] Ben Cao, Sue Zhao, Xue Li, and Bin Wang. K-means multi-verse optimizer (kmvo) algorithm to construct dna storage codes. *Ieee Access*, 8:29547–29556, 2020.

[4] Arturo Casini, Marko Storch, Geoffrey S Baldwin, and Tom Ellis. Bricks and blueprints: methods and standards for dna assembly. *Nature Reviews Molecular Cell Biology*, 16(9):568–576, 2015.

[5] Luis Ceze, Jeff Nivala, and Karin Strauss. Molecular digital data storage using dna. *Nature Reviews Genetics*, 20(8):456–466, 2019.

[6] Ran Chao, Yongbo Yuan, and Huimin Zhao. Recent advances in dna assembly technologies. *FEMS yeast research*, 15(1):1, 2015.

[7] Gadi Chaykin, Nili Furman, Omer Sabary, Dvir Ben-Shabat, and Eitan Yaakobi. Dna-storalator: end-to-end dna storage simulator. In *13th Annual Non-Volatile Memories Workshop*, 2022.

[8] Yeow Meng Chee and San Ling. Improved lower bounds for constant gc-content dna codes. *IEEE Transactions on Information Theory*, 54(1):391–394, 2008.

[9] George M Church, Yuan Gao, and Sriram Kosuri. Next-generation digital information storage in dna. *Science*, 337(6102):1628–1628, 2012.

[10] Delphine Coudy, Marthe Colotte, Aurélie Luis, Sophie Tuffet, and Jacques Bonnet. Long term conservation of dna at ambient temperature. implications for dna data storage. *PLoS One*, 16(11):e0259868, 2021.

[11] Douglas Densmore, Timothy H-C Hsiau, Joshua T Kittleson, Will DeLoache, Christopher Batten, and J Christopher Anderson. Algorithms for automated dna assembly. *Nucleic acids research*, 38(8):2607–2616, 2010.

[12] Yiming Dong, Fajia Sun, Zhi Ping, Qi Ouyang, and Long Qian. Dna storage: research landscape and future prospects. *National Science Review*, 7(6):1092–1107, 2020.

[13] Danny Dubé, Wentu Song, and Kui Cai. Dna codes with run-length limitation and knuth-like balancing of the gc contents. In *Symposium on Information Theory and its Applications (SITA), Japan*, 2019.

[14] Robert E Fontana and Gary M Decad. Moore's law realities for recording systems and memory storage components: Hdd, tape, nand, and optical. *AIP Advances*, 8(5), 2018.

[15] Philippe Gaborit and Oliver D King. Linear constructions for dna codes. *Theoretical computer science*, 334(1-3):99–113, 2005.

[16] Andreas L Gimpel, Wendelin J Stark, Reinhard Heckel, and Robert N Grass. A digital twin for dna data storage based on comprehensive quantification of errors and biases. *Nature Communications*, 14(1):6026, 2023.

[17] Nick Goldman, Paul Bertone, Siyuan Chen, Christophe Dessimoz, Emily M LeProust, Botond Sipos, and Ewan Birney. Towards practical, high-capacity, low-maintenance information storage in synthesized dna. *nature*, 494(7435):77–80, 2013.

[18] John E Gorzynski, Sneha D Goenka, Kishwar Shafin, Tanner D Jensen, Dianna G Fisk, Megan E Grove, Elizabeth Spiteri, Trevor Pesout, Jean Monlong, Gunjan Baid, et al. Ultrarapid nanopore genome sequencing in a critical care setting. *New England Journal of Medicine*, 386(7):700–702, 2022.

[19] Kees A. Schouhamer Immink and Kui Cai. Properties and constructions of constrained codes for dna-based data storage, 2018.

[20] Kees A Schouhamer Immink and Kui Cai. Efficient balanced and maximum homopolymer-run restricted block codes for dna-based data storage. *IEEE Communications Letters*, 23(10):1676–1679, 2019.

[21] Yuxi Ke, Eesha Sharma, Hannah K. Wayment-Steele, Winston R. Becker, Anthony Ho, Emil Marklund, and William J. Greenleaf. High-throughput dna melt measurements enable improved models of dna folding thermodynamics. *bioRxiv*, 2024.

[22] Dixita Limbachiya, Manish K. Gupta, and Vaneet Aggarwal. Family of constrained codes for archival dna data storage. *IEEE Communications Letters*, 22(10):1972–1975, 2018.

[23] Hannah F Löchel, Marius Welzel, Georges Hattab, Anne-Christin Hauschild, and Dominik Heider. Fractal construction of constrained code words for dna storage systems. *Nucleic Acids Research*, 50(5):e30–e30, 2022.

[24] Karishma Matange, James M Tuck, and Albert J Keung. Dna stability: a central design consideration for dna data storage systems. *Nature communications*, 12(1):1–9, 2021.

[25] Linda C Meiser, Philipp L Antkowiak, Julian Koch, Weida D Chen, A Xavier Kohll, Wendelin J Stark, Reinhard Heckel, and Robert N Grass. Reading and writing digital data in dna. *Nature protocols*, 15(1):86–101, 2020.

[26] André E Minoche, Juliane C Dohm, and Heinz Himmelbauer. Evaluation of genomic high-throughput sequencing data generated on illumina hiseq and genome analyzer systems. *Genome biology*, 12(11):1–15, 2011.

[27] Inbal Preuss, Michael Rosenberg, Zohar Yakhini, and Leon Anavy. Efficient dna-based data storage using shortmer combinatorial encoding. *Scientific reports*, 14(1):7731, 2024.

[28] Michael Schwarz, Marius Welzel, Tolganay Kabdullayeva, Anke Becker, Bernd Freisleben, and Dominik Heider. Mesa: automated assessment of synthetic dna fragments and simulation of dna synthesis, storage, sequencing and pcr errors. *Bioinformatics*, 36(11):3322–3326, 2020.

[29] Wentu Song, Kui Cai, Mu Zhang, and Chau Yuen. Codes with run-length and gc-content constraints for dna-based data storage. *IEEE Communications Letters*, 22(10):2004–2007, 2018.

[30] Yanfeng Wang, Yongpeng Shen, Xuncai Zhang, and Guangzhao Cui. Dna codewords design using the improved nsga-ii algorithms. In *2009 Fourth International on Conference on Bio-Inspired Computing*, pages 1–5. IEEE, 2009.

[31] Yixin Wang, Md. Noor-A-Rahim, Erry Gunawan, Yong Liang Guan, and Chueh Loo Poh. Construction of bio-constrained code for dna data storage. *IEEE Communications Letters*, 23(6):963–966, 2019.

6 Supplementary Material

6.1 On the construction of complete libraries for half-overlapping ligation schemes

The following lemma offers a simple approach to constructing complete libraries from left and right motif sets. Let us start with the following definitions:

Lemma 6.1. Let $L,R \subseteq \mathscr{A}_{n/2}$ denote the *set of left and right motifs*, respectively. Further, let the function c concatenate the left and right halfs of an oligo of length n, $c : L \times R \mapsto \mathscr{A}_n : c(l,r) = (l_1,\dots,l_{n_l},r_1,\dots,r_{n_r})$. Then, the set

$$\mathscr{A}(L,R) := c((L \cup L^c) \otimes (R \cup R^c)) \tag{4}$$

is a complete library, where $\otimes$ denotes the cartesian product of two sets.

Proof. The show the completeness of the library, we need to argue that for every pair of oligos o and p in $\mathscr{A}(L,R)$, there exists an oligo l in $\mathscr{A}(L,R)$ that can hybridise to the right motif of o and the left motif of p, without loss of generality. Thus, we need to show that for every motif o_r in $R \cup R^c$ and p_l in $L \cup L^c$, the reverse complements o_l^c and p_r^c are also in their respective sets. This is the case, since the reverse complementarity operator is involutory and, thus, the reverse complement of the reverse complement of a motif is the motif itself. Thus, l in $L \cup L^c$ and $R \cup R^c$ are closed under the reverse complementarity operator and since every combination of left and right halfs is in the cartesian product of these sets, the library is complete. □

Note, that if L has n_L pairs and R has n_R pairs, the sets consist of $2n_L$ and $2n_R$ oligos, respectively. The only exeption to this rule are symmetric motifs where the reverse complement of the motif is the motif itself. In this situation the oligo pair consists of a single element. The cartesian product of L and R (every combination of left and right halfs), consists of $4n_Ln_R$ elements. For example, eight symbols on the left and right, ($n_L = n_R = 8$), will lead to a library with a cardinality of 256. Note, that it is easiest to find simple encoding functions if $|L| = |R| = 2^k$ for some k because this allows to directly base it on the binary representation of a number, see 2. Also, among all encoding libraries with a given number of *symbols* $N = |L| + |R|$ the library with the greatest density is the one where $|L| = |R| = \frac{N}{2}$.

6.2 Calculation of the expected number of pooling steps in a serial ligation scheme

Let L_i denote the random variable that represents the ith symbol of the left side of R_i as the ith symbol on the right side of an oligo. A complete codeword is made up of a sequence of such random variables. Consider e.g., codewords of length 256. Then the codeword consists of a sequence $(L_1,R_1,L_2,R_2,\dots,L_{32},R_{32})$. We can assume, that hese random variables are independent and identically distributed and follow a uniform distribution on the respective sets. $P[L_i = S] = 1/|L| \quad \forall S \in L$ and the same holds true for R_i. Note, that the ligation oligos are determined by the message oligos and are, thus, no random variables in this context.

We can now formulate the conditions for every random variable L_i and R_i such that $L_1,\dots,L_i$ and $R_1,\dots,R_i$ represent a poolable set of oligos. Lets begin with the case, where $M = L$, were all elements are used both as message and ligation oligos.

- $L_1 \in L$

- $R_1 \in R$
- $L_2 \in L \setminus \{L_1, L_1^{'}\}$
- $R_2 \in R \setminus \{R_1, R_1^{'}\}$
- $L_3 \in L \setminus \{L_1, L_1^{'}, L_2, L_2^{'}\}$
- $R_3 \in R \setminus \{R_1, R_1^{'}, R_2, R_2^{'}\}$
- $\ldots$

The probability that the first i oligos are poolable is, thus, given by

$$P[((L_1,R_1),\ldots,(L_i,R_i)) \text{ are poolable}] = \prod_{j=1}^{i-1}(1-\frac{2j}{|L|})(1-\frac{2j}{|R|}) \quad \forall i = 1,\ldots,\min(|L|,|R|) \tag{5}$$

If X denotes the random variable that represents the length of the longest poolable set of oligos, then

$$X = i \iff ((L_1,R_1),\ldots,(L_i,R_i)) \text{ are poolable} \wedge (L_1,R_1,\ldots,L_{i+1},R_{i+1}) \text{ are not poolable.} \tag{6}$$

$$P[X=i] = \prod_{j=0}^{i}(1-\frac{2j}{|L|})(1-\frac{2j}{|R|})(\frac{2(i+1)}{|L|}+\frac{2(i+1)}{|R|}-\frac{2(i+1)}{|L|}\frac{2(i+1)}{|R|}) \quad \forall i = 1,\ldots,\min(|L|,|R|) \tag{7}$$

For the case, where $|L|=|R|$ this simplifies to

$$P[X=i] = \prod_{j=1}^{i}(1-\frac{2(j-1)}{|L|})^2(\frac{4i}{|L|}-\frac{4i^2}{|L|^2}) \quad \forall i = 1,\ldots,|L|. \tag{8}$$

The expected value $\mathbb{E}[X]$ is then given by:

$$\mathbb{E}[X] = \sum_{i=1}^{|L|/2} i \cdot \prod_{j=1}^{i}(1-\frac{2(j-1)}{|L|})^2(\frac{4i}{|L|}-\frac{4i^2}{|L|^2}) \tag{9}$$

The most naive approach is to start from the beginning of the sequence and to pool as many oligos as possible and continue in this manner. There obviously are more efficient approaches, but this is an upper bound for the number of ligation reactions required to assemble a codeword. If m is the codeword length and n is the number of nucleotides in an oligo, then the number of oligos per codeword is m/n. To obtain the number of pools required to assemble the first , we can simple divide by the expected number of oligos per pool $\mathbb{E}[X]$. The number of first level pooling reactions is, thus,

$$\frac{m/n}{\mathbb{E}[X]} \tag{10}$$

The resulting double stranded oligos need to further be ligated with each other on the second pooling level, then the third etc. yielding a total number of

$$\sum_{i=1}\frac{m/n}{\mathbb{E}[X]^i} \leq \frac{m}{n}\frac{1}{1-\frac{1}{\mathbb{E}[X]}} = \frac{m}{n}\frac{\mathbb{E}[X]}{\mathbb{E}[X]-1} \tag{11}$$

Following the reasoning above, there is a tradeoff between the size of the library and the number of ligation steps required to assemble a codeword. In fact, the largest number of oligos that can be pooled in a single pool is $\min(|L|,|R|)$. The objective of a pooling algorithm must be to identify subsets the oligos of a codeword, that do not contain duplicate symbols. Since the symbols are drawn randomly, this will rarely be possible.

6.3 Pooling Algorithm

Let $L_1, R_1, \ldots, L_{m/n}, R_{m/n}$ denote the sequence of symbols of a codeword to be synthesised.

1. The initial pool consists of all oligos in the sequence. $P_0 := \{L_1, R_1, \ldots, L_{m/n}, R_{m/n}\}$
2. For each symbol $l \in L$ and $r \in R$, count the number of occurences of l and r in the first i symbols of the codeword. This yields $|L|+|R|$ different vectors of length $m/2n$.

   E.g. if the symbol l_1 occurs on the indices 4,5,7,11,12,14, then the vector c_{l_1} will be
3. Select the symbol that has the closest two occurences of the same symbol and break the pool into two pools at that particular position. In our example this would be at position 12.
4. Define new pools as $P_1 = \{L_1, R_1, L_{11}, R_{11}\}$ and $P_2 = \{L_{12}, R_{12}, \ldots, L_{m/n}, R_{m/n}\}$.
5. Repeat steps 2-4 until every symbol only appears not more than once in any pool.

This algorithm will not yield the minimal number of pools, but can easily be improved by joining several pools that do not contain duplicate symbols. Asymptotically, as the length of codewords increases, the expected number of oligos per pool will increase to the number of different symbols in the library.

6.4 Binomial Encoding

Let us denote the cost of assembling a single oligo by c. If the cost of assembling k oligos is also c, then the binomial encoding approach significantly reduces the costs per bit. If, however, the cost of assembling k oligos is $k \cdot c$, then the binomial encoding approach does not reduce the costs per bit, as can be seen in the right part of Figure 12.

Message oligos, which encode the actual information have no motive restriction and are therefore depicted empty. Positional oligos are depicted with a motive, which determines at what position the double standed oligo is supposed to be ligated. This strategy enables both the parallel assembly of different variants of a codeword in a single pool as well as the assembly of all positional oligos in parallel and, thus, reduces, the pooling levels to two. Since our library consists of 16 left and 16 right motives, then a codeword can contain 32 different positions and the assembly of a codeword only requires 33 pools. Note, that this approach also requires the use of a barcode for every oligo in order to identify the strad variant. This slightly reduces the data density of the encoding scheme, depending on the kind of barcode used.

In a real application, the optimal k will be determined by the copy numbers of the assembly process and can, thus, be considered a given parameter for the problem of designing an optimal binomial encoding schema. However, from the binomial function it is clear that the most information is encoded when k is $\lfloor n/2 \rfloor$. This is depicted on the left side of Figure **??**. The right side of the figure shows the diminishing return to the library size if k is fixed (at 5). We must, thus, attempt to set k as close to $\lfloor n/2 \rfloor$ as possible.

In order to obtain unambiguous hybridisations is the specified pooling schema, there are two constraints that need to be satisfied. Given a motive of A at position i in a codeword, the motive at position $i+2$ must not be A or A'. If these two conditions are satisfied, parallel assembly of all variants of a block can be achieved.

Instead of constraining the sequence, however, we can also constrain the *Sigma* space. Consider a codeword with motive A on position i, then the following oligo with motives X and Y can be selected, from 16 and 14 different motives, effectively reducing the number of possible oligos from the library to 256 - 32 = 224.

6.5 Thermodynamic properties

However, the thermodynamic properties of the sequences that compose the motifs on a library are crucial for unambigous hybridisation and ligation as well as avoiding secondary structures and binding issues in general.

There are many problems to be considered. For sequencing effectiveness, it is essential that

There are essentially two requirements for an encoding scheme to satisfy the no-homopolymer condition. Firstly, the library of oligos has to be constructed in such a way that every oligo in the library satisfies the no-homopolymer condition. Second, the ligation of two oligos must not create homopolymers between oligos.

There are two ways to ensure the second condition. The first approach is to select a library of oligos in such a way, that homopolymers can never occur, irrespective of the order in which the oligos are ligated. A straightforward approach would be to impose conditions of the first and the last nucleotide, e.g. ($\forall o \in \mathscr{A} : o_1 \in \{A,C\} \wedge o_n \in \{G,T\}$). Any series of oligos could then be concatenated to a no homopolymer DNA sequence. Furthermore, the condition also imposes a condition on the $n/2$th and $n/2+1$th nucleotide of ligation oligos. By inverse complementarity, ($\forall o \in \mathscr{A} : o_{l/2} \in \{G,T\} \wedge o_{l/2+1} \in \{A,C\}$). If ligation oligos also start with A or C, then the condition also holds for coding oligos. If, however, ligation oligos start with G or T and end with A or C, the library would be split into disjoint sets: ($\mathscr{A} = \mathscr{O} \cup \mathscr{I}$, where $\mathscr{O} \cap \mathscr{I} = \emptyset$).

The simulation created 25 libraries with n bases per oligo. These have oligos with around 50% CG-content where the amount of variation of the CG-content is unique to every oligo length. Further, the oligos were picked such that the intended hybridization has energies above the median of all possible binding energies for $n/2$ Watson-Crick bindings.

The values analyzed here are the Medians of intended and unintended bindings $\bar{x}$, the median absolute deviation (MAD) MAD $=$ median$(|x_i - \bar{x}|)$, the range of ΔG values for both intended and unintended bindings $\max(A) - \min(A)$ where A is the set of intended or unintended bindings, the gap size $|\max(I)| - |\min(U)|$ where I is the set of intended bindings and U is the set of unintended bindings, and lastly the amount of unintended bindings with high energy $|\{y | y \in U, y > \min(I)\}|$.

Many constrained encoding schemes are designed to have a balanced CG content. [23, 19, 29, 20, 31, 13, 22, 30, 3, 15, 8]. This condition is, however, only a simplification of the thermodynamic properties of DNA. More general approaches require the Gibbs free energy of the oligos to be within a certain range....

6.6 Experimental Validation

6.6.1 Melting curve measurements: Materials and Methods

The library of 5'-phsophorylated 20-nt oligos has been ordered from IDT. Melting curve measurements have been performed with the Roche Light Cycler® 480 II using 96 well semi-skirted plates (FrameStar, # 4TI-0952) and adhesive seal (4titude, # 016540F). Each 20µL reaction contained 0,5µM of each two oligos, 1x EvaGreen® (# 31000), 1mM MgCl2 and nulease free water. Of all possible combinations to asses a RCP, only one was arbitrarily chosen and reactions were prepared in duplictaes. The melting curve measurement was performed with an initial heating step to 95 °C (ramp rate 4,4 °C/s; hold 5 sec), followed by cooling to 20 °C (ramp rate 2,2 °C/s, hold 1 min) and continous heating to 95 °C at a ramp rate of 0,11 °C/s and 5 acquisitions per °C. Tm calling was performed with the Light Cycler® 480 software (v1.5.1) and the melting curves were analysed and plotted using Microsoft® Excel (16.89.1). Melting temperatures of RCP were predicted using IDT's OligoAnalyzer™ Tool using the following parameters: Target Type (DNA), Oligo Conc (0.25; because if [strand 1] = [strand2], Oligo Conc = ([strand1]+[strand2])/4, according to "melting temperature assumptions and limitations")), Na+ Conc (0mM), Mg++ (2,5mM), dNTPs Conc (0mM).

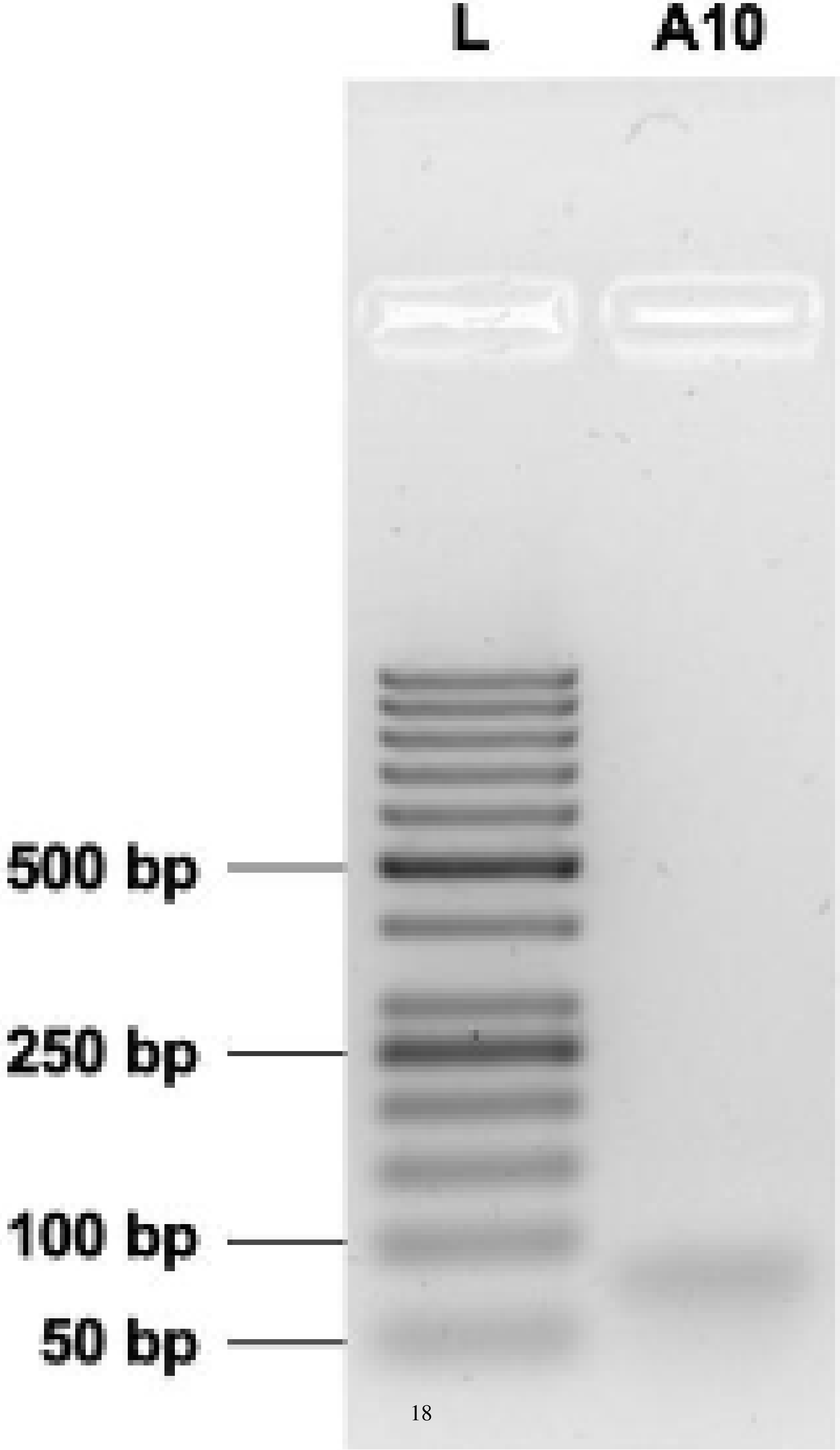
6.6.2 Ligation Experiments: Materials and Methods

Unless indicated otherwise all ligases are sourced from New England Biolabs. A 256-oligonucleotide library of 20-mers was sourced from IDT. The library was carefully designed based on the algorithm decribed in 6.1. Original source library concentration comes at 100 μM in nuclease free water. Oligo library was then diluted at different working concentrations of 10 μM, 1 μM and 0,1 μM with nuclease free water. Ligation reactions take place in 20 μl final volume and are prepared as following: 1 μl of each oligo is pipetted according to the computer-generated pooling scheme in 2μl of 10X ligase buffer and nuclease free water. Finally, 1 μl of ligase is added to the mix. The reaction occurs for 30 min at 25 °C. As negative control, ligation reaction was prepared the same way except that oligonucleotides were replaced by nuclease free water.

Analysis of ligation products with agarose gel electrophoresis was conducted using 2% Agarose (Avantor VWR, USA) + 1X Tris-Acetate-EDTA buffer (Avantor VWR, USA) gel was prepared, and stained with SYBR Safe (Thermo Fisher Scientific). For Gel loading, 8 µl of ligation product was mixed with 2 µl of Gel loading Dye, Purple 6X (New England Biolabs) and 2 µl of nuclease free water. Ladders used in this work are sourced from Invitrogen (Thermo Fisher Scientific). Gel was migrated at 90 V for 2h. Pictures of the gels were taken with a FastGene FAS-X (Nippon genetics, Germany) imager system.

The analysis of the ligation products were performed on a 5300 fragment analyser (Agilent) with a 1 - 6000 pb DNF-974-33 kit. In capillary electrophoresis parameters: Sample was first injected at 5.0 kV for 30 s and separation occurred at 6.0 kV for 50 min. Electropherograms obtained were analysed with PROSize 3.0 software.

As a first proof of concept, we tested whether ligation could occur in free solution without any surface attachment (e.g. magnetic beads or glass surface) (Pengpumkiat et al, 2016; Barisic et al, 2015). For this, a pool forming a fragment longer than 50 base pairs, containing unique motives was selected from the pipetting scheme. This pool (poolA10) consists of 8 oligonucleotides and is expected to form an 80-bp fragment upon correct annealing and ligation reaction. Ligation reaction was prepared as mentioned in Supplementary Material 6.6 with T4 DNA ligase (20 U/µl final). In these conditions a band at approximately 80 pb (expected size of pool A10) was observed on Agarose + 1X TAE gel (Fig. 2). No bands were observed in control conditions (not shown here) where oligos are pooled together without adjunction of Ligase nor in conditions of ligase alone. Additionally not differences were observed when oligos are pooled together in nuclease free water or in T4 ligase buffer.


L
A10
500 bp
250 bp
100 bp
50 bp

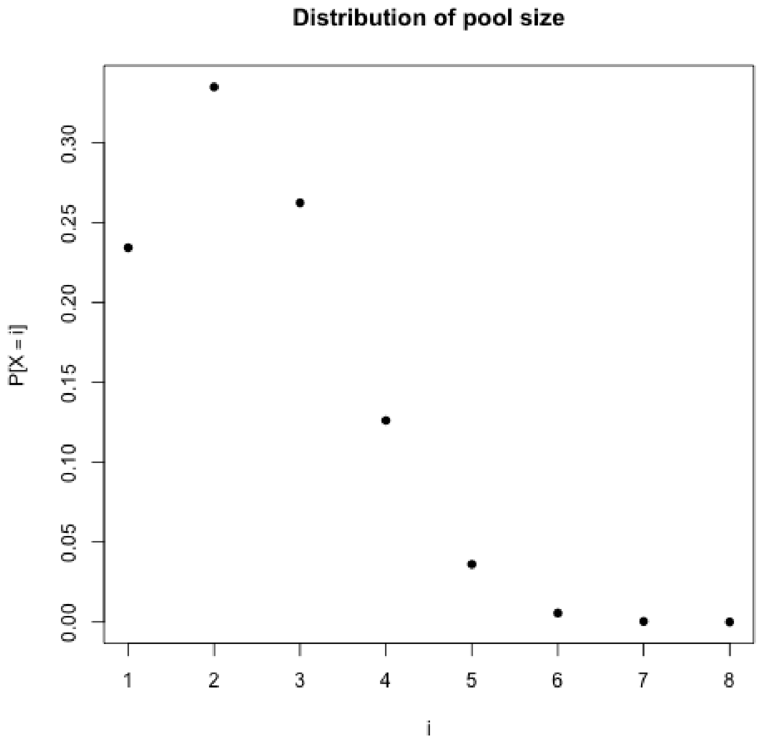


Figure 10: The distribution of the number of poolable oligos for $|L|=|R|=16$ and $M=L$. The expected value is 2.41.

Figure 11: The expected pool size and and upper bound for the number of pools depending on the library size.

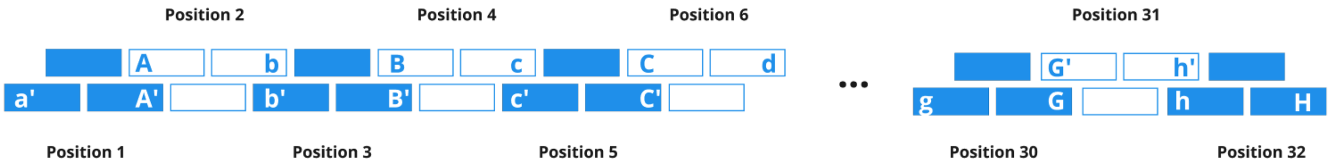


Figure 12: (Left)A comparison of the data encoded in a single position of a normal encoding scheme and a binomial encoding scheme. (Right) The cost of encoding a single bit expressed as number of assembled oligos for different assembly methods. For both illustrations, we fixed $k=5$.

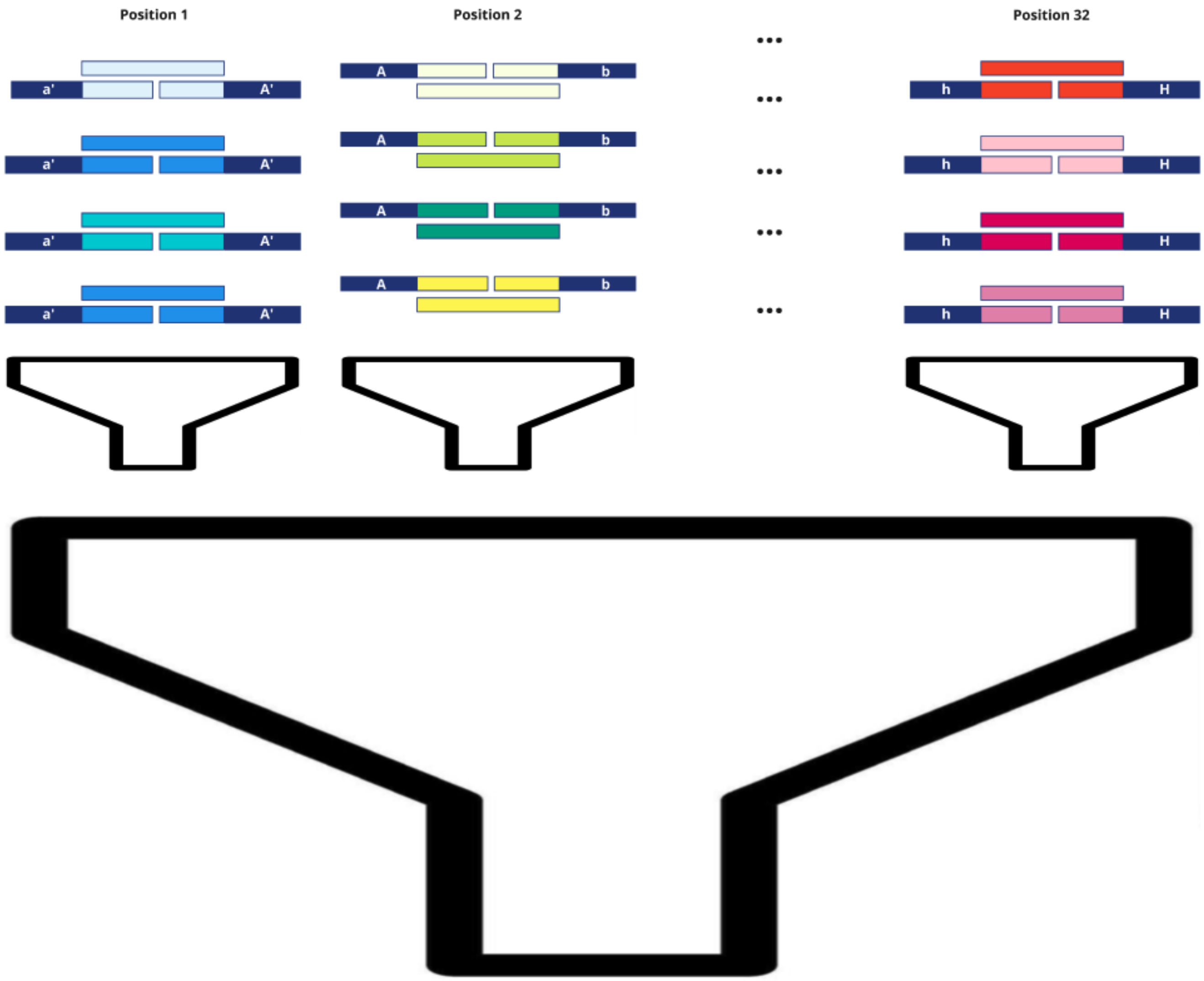


Figure 13: An illustration of the binomial encoding schema, consisting of message oligos and positional oligos.

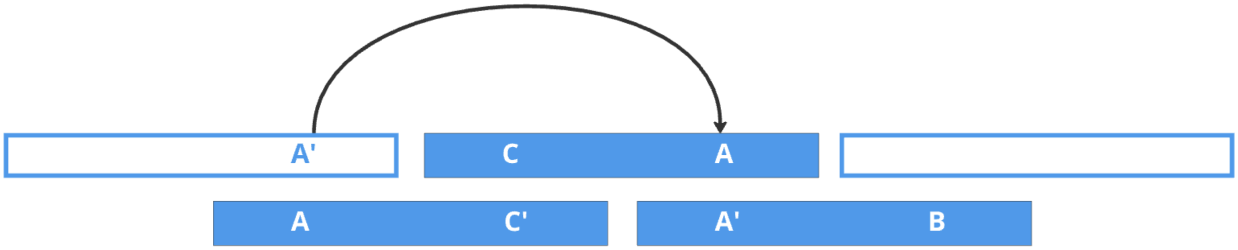


Figure 14: An illustration of the constraints on the motive sequence.