The Capacity of DNA for Information Encoding

Vinhthuy Phan  Max H. Garzon

Computer Science, The University of Memphis, Memphis, TN 38152-3240, U.S.A.
{vphan, mgarzon}@memphis.edu

Abstract. Lower bounds on the information encoding capacity of DNA are established using a combinatorial approximation of the Gibbs energy landscapes using the so-called $h$-distance as a model of DNA homology. These bounds decrease exponentially with a parameter $\tau$ that roughly codes for stringency in reaction conditions. Second, we introduce a family of codes, shuffle codes, that are provably almost as large as optimal codes. This shuffling construction can be used to produce codes with a constant GC-content. These close-to-optimal codes yield estimates of the capacity of DNA oligonucleotides to store biotic information in DNA arrays as defined in [13]. Finally, we discuss the capacity of DNA arrays as a mechanism to store distinguishable information.

Keywords Information storage capacity of DNA, codeword design, tensor product methods, Gibbs energy, $h$-distance.

1 Introduction

The ways cells and biomaterials store and process information is one of the most important and enigmatic problems of our times. One major subproblem of this puzzle, encoding information in biomolecules for information processing in vitro, has proven to be a challenging and interesting problem for biomolecule-based computing after Adelman’s founding work [1]. Codeword design and, more generally, data and information representation in DNA bear an increasing interest, not only from the point of view of using biomolecules for computation, but also for shedding light on a number of other problems in areas outside computation per se, such as bioinformatics, and conceivable microbiology and genetics. The codeword design problem [13, 8, 5, 10, 16, 15, 4] requires producing sets of strands that are likely to bind in desirable hybridizations while minimizing the probability of erroneous hybridizations that may induce false positive outcomes. A fairly extensive literature now exists on various aspects and approaches of the problem (see [5] for a review.) Although various algorithms have been proposed for testing the quality of codeword sets in terms of being free of secondary structure [5, 9], very few methods have been proposed to systematically produce codes of high enough quality to guarantee good performance in test tube protocols [12]. Even fewer articles address the issue of the capacity of DNA strands of a given length (say 20-mers) to hold information, if they are to co-exist without interactions in a test tube under given stringency conditions, such as required by an associative memory "in principle larger than the brain" [4].
The obvious approach to encode information in DNA is to encode symbolic strings in DNA strands. Direct encoding is, however, not very efficient for storage or processing of massive amounts (over terabytes) of abiotic data because of the enormous implicit cost of DNA synthesis to produce the encoding sequences. Indirect and more efficient methods have been proposed [12,14], assuming the existence of a large basis of non-cross-hybridizing DNA molecules, as estimated in this paper. DNA molecules can interact through intermolecular reactions, usually hybridization in DNA form alone. Due to the inherent uncertainty in biochemical processes, small variations in strand composition will not cause major changes in hybridization events, with consequent limitations on using similar molecules to encode different records. Input strands must be “far apart” from one another in hybridization affinity in order to prevent disturbing cross-hybridizations. The major difficulty is that the hybridization affinity between DNA strand sets is difficult to quantify. Ideally, the Gibbs energy released in the process is the most appropriate criterion, but its exact calculation is difficult, even for pairwise interactions among small oligos, and using approximation models [7]. Hence, an exhaustive search of strand sets of words maximally separated in a given coding space (codeword sets) is infeasible, even for the small oligo-nucleotide sizes useful in DNA computing. In particular, the capacity of DNA oligonucleotides of a given length to store information appears to be very difficult to quantify, or even estimate.

To cope with this problem, a much simpler and computationally tractable model of the Gibbs energy landscapes was introduced in [15], using a metric for hybridization affinity known as the $h$-distance. As the $h$-distance does not assume an alignment between strands as is the case in simpler models [3, 19, 21], it is therefore applicable to situations in vitro, it has been demonstrated it produces results consistent with experiments in vitro of actual hybridizations in the range of 20- to 60-mers in PCR reactions [12, 6, 15, 16]. Since the $h$-distance model (described in Section 2) is more realistically restrictive than the Hamming distance model, two strands that are of distance $\tau$ apart in the $h$-distance model are also at least that much apart in the simpler model, and thus our constructed codes described in this paper are also codes in the simpler model.

Our results can be summarized as follows. Lower bounds on maximal codes, which are analogous to the the Gilbert-Varshamov sphere-packing bounds of Hamming codes, with $h$-distance $\tau$ are established; see Section 3. We propose a simple and innovative method to construct explicitly shuffle codes with and without the requirement of having a constant GC-content; see Section 4, 5. These code sizes exceed the established lower bounds, and are proven to be very close to optimal sizes when the stringency $\tau$ is not too large. Table 1 shows sizes of constructed codes from various practical lengths $n$ and two typical stringency of 100% and 50%. The last column shows the sizes of $(n, \frac{n}{2}, \frac{n}{2})$-code, whose GC-content is $\frac{n}{2}$. The established bounds also provide an estimate of the capacity of DNA arrays to store information by distributed representations. Finally, we discuss the capacity of DNA arrays as a mechanism to store distinguishable information; see Section 6.
|
|---|---|---|---|
| Length (n) | $\tau = n$ | $\tau = \frac{n}{2}$ | $t = \frac{n}{2}$, $w = \frac{n}{2}$ |
| 12 | 4,096 | 46,656 | 4,096 |
| 16 | 65,336 | 1,679,628 | 65,336 |
| 20 | 1,048,580 | 60,466,200 | 1,048,580 |
| 24 | 16,777,200 | 2,176,780,000 | 16,777,200 |
| 28 | 268,435,000 | 78,364,200,000 | 268,435,000 |

Table 1. Sizes of shuffle codes with stringency pre-assigned at 50% and 100%, respectively. The last column shows imposes an additional requirement of 50% of GC content.

## 2 Gibbs Energy and the $H$-measure

Ideally, the Gibbs energy released in the hybridization process between strand pairs is the most appropriate criterion of quality for a code set for experiments in-vitro [25]. Although hybridization reactions in vitro are governed by well-established rules of local interaction between base pairings, difficulties arise in trying to extend these rules to even short oligonucleotides (less than 150-mers) in a variety of conditions [24]. Hence, an exhaustive search of strand sets of words maximally separated in a given coding space is infeasible, even for the small size of oligo-nucleotides useful in DNA computing. Computation of the Gibbs energy thus relies on approximations based on various assumptions about the type of interactions between neighboring bonds. Various models have been proposed for the range of oligonucleotides used in DNA-based computing, major among which are the nearest-neighbor model and the staggered-zipper model [24]. An extension of the nearest neighbor model proposed by [7] computes optimal alignments between DNA oligonucleotides using a dynamic programming algorithm. Although perhaps acceptable as a model to gauge homology between two strands, it becomes prohibitively expensive when used to check a massive number of combinatorial possibilities among large codeword sets. To address this problem, a more computationally efficient concept, various metrics have been considered in the literature [3, 19, 21, 16] and various estimates have thus been given.

In the model using the $h$-measure, the Gibbs energy is approximated by a measure counts the maximally possible basepair matches in all possible alignments of two DNA strands. Let $x, y$ be two DNA sequences of length $n$ (written from the 5'- to the 3'-end.). Define

$$h(x, y) := \min_{-\frac{n}{2} \leq k < \frac{n}{2}} \{ |k| + H(x, \sigma^k(y^{wc})) \}$$

(1)

where $\sigma^k$ is the (right-) left-shift by $k$ positions (if $k < 0$, respectively.), $y^{wc}$ is the Watson-Crick-complement of $y$ obtained by reversing $y$ and exchanging A's for T's and vice versa, and C's for G's and vice versa, and $H(\ast, \ast)$ is the ordinary Hamming distance. The $h$-measure considers hybridization in all possible frame-shifts, which is more realistically restrictive than simpler models [19, 21] in which
hybridization is considered only in the perfect alignment (i.e., shift 0). Measure 0 indicates perfect complementarity. A large measure indicates that even when $x$ finds itself in the proximity of $y$, they contain few complementary basepairs, and are less likely to hybridize. This measure $h$ can be precisely related to the maximum number of complementary base pairs in all frame shifts as follows:

**Lemma 1.**

$$h(x, y) = n - \max_{-(n-1) \leq i \leq n-1} \{m(x, \sigma^i(y^{wc}))\}$$

where $m(x, \sigma^i(y^{wc}))$ is the number of matches of $x$ and $y^{wc}$ in frame shift $i$.

**Proof.** Let $s$ be a shift in which $h(x, y) = \tau = |s| + H(u, \sigma^s(v^{wc}))$ is minimum. In shift $s$, only $n - |s|$ characters are aligned. Of these, some are complementary (measured by $m(u, \sigma^s(v^{wc}))$) and some are not (measured by $H(u, \sigma^s(v^{wc}))$). Therefore, $|s| + H(u, \sigma^s(v^{wc})) + m(u, \sigma^s(v^{wc})) = n$, and consequently, $h(x, y) = n - m(u, \sigma^s(v^{wc}))$. Further, since $s + H(u, \sigma^s(v^{wc}))$ is minimum over all shifts $s$, $m(u, \sigma^s(v^{wc}))$ is maximum over all shifts $s$. \ 

## 3 Bound on DNA Codes

To achieve effective DNA-based computation, the set of DNA strands involved in such computation should exhibit as few cross-hybridized pairs as possible other than the planned hybridizations between complementary segments. The $h$-measure only abstracts the likelihood of hybridization between arbitrary strands under ideal conditions. Therefore, the reaction conditions and stringency upon which hybridization may take place is also abstracted as an additional parameter $\tau$. A set of sequences $S$ is called an $(n, \tau)$-code if $\forall x, y \in S$, $h(x, y) \geq \tau$. Therefore, if we expect that no two strands whose $h$-measure is less than $\tau$ hybridize when they encounter each other, then we expect that the strands of an $(n, \tau)$-code are not cross-hybridizing under any circumstances. When the tube temperature is sufficiently high, perfectly complementary can barely hybridize, so the reaction conditions are very stringent. This corresponds to the $(n, 0)$-code, which contains all possible $4^n$ strands. On the other hand, an $(n, n)$-code has the least number of strands, which are required pairwise to be exactly $n$ from each other, i.e. no bases whatsoever can hybridize.

To analyze the size of a maximal $(n, \tau)$-code, we show an analogous result to bounds on maximal Hamming codes in a classical information theoretic context [22]. A Hamming code, $A(\tau)$, is bound in size by the Sphere-packing and Gilbert-Varshamov bounds as $\frac{V}{\alpha(\tau)} \leq |A(\tau)| \leq \frac{V}{\alpha(\tau) + 1}$, where $V$ is the number of all possible codewords, and $\alpha(\tau)$ is the number of all codewords of distance $\tau$ from any point. We will argue that an $(n, \tau)$-code in the space of DNA strands of length $n$ can be viewed as an $A(2)$ code in some metric space (to be defined), and similarly $\frac{V}{\alpha(2)} \leq |A(2)| \leq \frac{V}{\alpha(2) + 1}$. First, we need to construct a metric space as the $h$-measure is not a metric over the space of sequences. The $h$-measure, specifically, creates a non-uniformity that makes the volumes of codewords depends
heavily on their centers. First, let $G_{n,\tau}$ be a graph with $4^{n}$ vertices representing all possible strands of length $n$, and edges induced by the $h$-measure: $(u,v)$ is an edge if $h(u,v) < \tau$. Define $d(u,v)$ to be the length of the shortest path in $G_{n,\tau}$ between $u$ and $v$. Since three conditions $d(x,x) = 0, d(x,y) = d(y,x)$, and $d(x,y) + d(y,z) \geq d(x,z)$ are satisfied, $(d, G_{n,\tau})$ is a metric space.

Lemma 2. An independent set of $G_{n,\tau}$ is an $(n,\tau)$-code.

Proof. Since edges in $G_{n,\tau}$ indicate those strands which are expected to hybridize once they come under close contact, a set of strands which do not mutually hybridize (an independent set of $G_{n,\tau}$) form an $(n,\tau)$-code.

Theorem 1. Let $S$ be a maximum $(n,\tau)$-code. Then, $\frac{V}{4^{n}} \leq |S| \leq V$, where $V = 4^{n}$ and $E$ is the number of edges of $G_{n,\tau}$. In particular, $|S| \geq \frac{4^{n} - \tau + 1}{\tau(\tau - 1)}$.

Proof. Apply Jensen’s inequality [17], $\frac{\sum_{i=1}^{V} f(x_{i})}{\sum_{i=1}^{V} x_{i}} \geq f\left( \frac{\sum_{i=1}^{V} x_{i}}{\sum_{i=1}^{V} f(x_{i})} \right)$, to the convex function $f(x) = \frac{1}{x}$ with $x_{i} = 1 + d_{i}$, where $d_{i}$ is the degree of vertex $i$, we get $\sum_{i=1}^{V} \frac{x_{i}}{1 + d_{i}} \geq \frac{V^{2}}{4^{n} E}$. Caro’s theorem [2] provides the lower bound on the size of a maximal independent set, $\alpha(G_{n,\tau})$: $|S| = \alpha(G_{n,\tau}) \geq \sum_{i=1}^{V} \frac{1}{1 + d_{i}} \geq \frac{V}{4^{n} E}$.

To quantify this lower bound, we estimate the number of edges in $G_{n,\tau}$. Let $H_{uv}$ be the event that two strands $u$ and $v$ hybridize. Define $H_{uv} = 1$ if $h(u,v) < \tau$, and $H_{uv} = 0$ otherwise. Then, $P(H_{uv} = 1) = P(h(u,v) \leq \tau - 1) \leq \tau^{(n \cdot \tau)} p^{n-\tau+1} = \tau^{(n \cdot \tau)} p^{n-\tau+1}$. The expected degree of a vertex $u$ is $E[d_{u}] = E[d_{uv}] = \sum_{v} P(H_{uv}) = 4^{n} \tau(p^{n-\tau+1} = \tau \cdot 4^{n} \tau(p^{n-\tau+1}).$ Now, we can apply the lower-bound result and get $|S| = \alpha(G_{n,\tau}) \geq \frac{4^{n}}{1 + 4^{n} \tau(p^{n-\tau+1})} \approx \frac{4^{n} - \tau + 1}{\tau(\tau - 1)}$. $\square$

Under the metric $d$ defined as above, the distance between two adjacent vertices of $G_{n,\tau}$ is 1. Inspite of the non-uniformity of $G_{n,\tau}$, the expected volume of the ball with radius 1 in this metric space, $Vol(1)$, is approximately $1 + \frac{2\tau}{4^{n}}$. In other words, the number of vertices (strands) of distance 1 away from a vertex is 1 (vertex itself) plus the average degree of $G_{n,\tau}$, which is $\frac{2\tau}{4^{n}}$. Under the same interpretation, $Vol(0)$ is 1 (the vertex itself). Furthermore, in this metric space the maximum independent set can be viewed a Hamming code of distance 2, or $A(2)$. Our result, then, implies that $\frac{V}{Vol(0)} \leq |A(2)| \leq \frac{V}{Vol(1)}$, which generalizes the Sphere-packing and Gilbert-Varshamov bounds.

4 Construction of Shuffle Codes

We now describe a method of constructing $(n,\tau)$-codes, given that $\tau$ must be a divisor of $n$. This construction yields a larger code than those established in Section 3. An explicitly construction has practical significance as these codes can be used in actual applications in larger biomolecular computing experiments and for the design of associative memories as discussed in Section 6. Briefly, our
method constructs a \((n, \tau)\)-code by \textit{shuffling} a \((\frac{n}{\tau}, 1)\)-code exactly \(\tau\) times. Given \(\{x_1, x_2, \ldots, x_k\}\), a set of DNA strands of length \(m\), define:

\[
x_1 \otimes \cdots \otimes x_k = x_11 \cdots x_k1 : x_12 \cdots x_k2 : \cdots : x_{1m} \cdots x_{km}
\]

where \(x_{1j}\) is the \(j^{th}\) character of the sequence \(x_1\), and so on. For example, for \(a_i, b_i, c_i, d_i \in \{A, C, G, T\}, 1 \leq i \leq 3\), \(a_1a_2a_3 \otimes b_1b_2b_3 \otimes c_1c_2c_3 \otimes d_1d_2d_3\) results in \(a_1b_1c_1d_1a_2b_2c_2d_2a_3b_3c_3d_3\). We now describe the algorithm that produces shuffle codes, absolute sizes of the produced codes, relative sizes of produced codes in comparison to optimal ones, and the efficiency of the algorithm.

**Algorithm 1** Input: \(n = \tau m\). Output: an \((n, \tau)\)-code

Construct \(S_m\), the set of all possible DNA strands of length \(m\).

\[
\text{for all strand } x \in S_m \text{ do}
\]

Construct \(x^c\), the unique complementary of \(x\), i.e. \(h(x, x^c) = 0\).

\[
S_m = S_m - x^c
\]

\[
\text{end for}[\text{note: } |S_m| = \frac{4^{m-|P_m|}}{2}]
\]

Let \(S = \{x_1 \otimes x_2 \otimes \cdots \otimes x_\tau : \forall x_i \in S_m \text{ and } x_i \neq x_j \text{ if } i \neq j\}\)

Return \(S\).

**Theorem 2.** Suppose \(n = \tau m\), then Algorithm 1 produces an \((n, \tau)\)-code of size

\[
\left(\frac{4^m - |P_m|}{2}\right)^\tau
\]

where \(|P_m| = |\{x : h(x, x) = 0, |x| = m\}| = 4^m\) if \(m\) is even, and 0 if \(m\) is odd.

**Proof.** First, we want to show that \(|S_m| = \frac{4^m - |P_m|}{2}\), where \(|P_m| = |\{x : h(x, x) = 0, |x| = m\}| = 4^m\) if \(m\) is even, and 0 if \(m\) is odd. To see that notice an \((m, 1)\)-code consists of no strand in \(P_m\). Further, for each strand \(x \in 4^m - P_m\), there is exactly one strand \(x^c \in 4^m - P_m\) such that \(h(x, x^c) = 0\). Removing those from \(4^m - P_m\), we get an \((m, 1)\)-code whose size is \(\frac{4^m - |P_m|}{2}\). Note that we only need to consider shift 0 because if \(h(x, y) = 0\), then their best alignment occurs in shift 0.

We now claim the set \(S\) produced by Algorithm 1 has \(\left(\frac{4^m - |P_m|}{2}\right)^\tau\) strands, because the shuffling of each unique set \(\{x_1, x_2, \ldots, x_\tau\}\), \(x_i \in S_m\) produces a unique strand \(x \in S\). Further, we claim that \(S\) is an \((n, \tau)\)-code; in other words, for any two set \(\{x_1, x_2, \ldots, x_\tau\}\) and \(\{y_1, y_2, \ldots, y_\tau\}\), \(x_i, y_i \in S_m\), \(h(x_1 \otimes \cdots \otimes x_\tau, y_1 \otimes \cdots \otimes y_\tau) \geq \tau\). The reason for this is because in any shift of the two shuffled strands, each \(x_i\) is perfectly aligned to some \(y_i^c\).

As an example, consider the alignment of \(a_1a_2a_3 \otimes b_1b_2b_3 \otimes c_1c_2c_3\) to \(x_1x_2x_3 \otimes y_1y_2y_3 \otimes z_1z_2z_3\). In any shift, strand \(a\) can be considered \textit{separately aligned} to either strand \(x\), or \(y\), or \(z\). The same holds for strands \(b\) and \(c\) Therefore, \(h(x_1 \otimes \cdots \otimes x_\tau, y_1 \otimes \cdots \otimes y_\tau) \geq \tau \cdot h(x_i, y_j) \geq \tau \cdot 1 = \tau\).
As a corollary, the algorithm constructs a provably maximal \((n, n)\)-code with 
\(\left(\frac{4^{1/2}}{2}\right)^n = 2^n\). In general, the shuffle code is greater than \(\frac{1}{2^n}\) of optimality.

**Theorem 3.** Let \(n = \tau m\) and \(L_\tau\) be a maximal \((n, \tau)\)-code, and \(S\) be the \((n, \tau)\)-

code constructed by Algorithm 1. If \(n\) or \(m\) is odd, then
\[
2^{\tau-1} \cdot |S| \geq |L_\tau|
\]

**Proof.** Let \(L_1\) be the maximal \((n, 1)\)-code. Then \(L_1 \geq L_\tau\), and thus \(\frac{|S|}{|L_1|} \leq \frac{|S|}{|L_\tau|}\).

Therefore,
\[
\frac{1}{2^{\tau-1}} \leq \frac{\left(4^n - |P_n|\right)^\tau}{2^\tau} \leq \frac{4^n - |P_n|}{2} = \frac{|S|}{|L_1|} \leq \frac{|S|}{|L_\tau|}
\]

which yields the result. Note that the first inequality holds because \(1 \leq \frac{4^n - |P_n|}{2} \leq \frac{4^n - |P_n|}{2}\) when \(n\) is odd or \(m\) is odd. \(\Box\)

**Theorem 4.** Algorithm 1 takes \(O(n \cdot |S|)\) steps and uses \(O(n \cdot |S|)\) memory, where \(S\) is the \((n, \tau)\)-code returned by the algorithm.

**Proof.** Note that \(n = \tau m\). Constructing \(S_m\) takes \(m \cdot 4^n\) steps. This involves going through all \(4^n\) sequences, each of which is excluded (in a \(O(m)\)-
step check) if its reverse complement is lexicographically equal or smaller than it. Constructing \(S = S_m\) requires \(|S| = |S_m|\) interleavings, each of which takes \(O(m \cdot \tau) = O(n)\) steps. Therefore, the total number of steps is \(O(m \cdot 4^n + n \cdot |S|)\).

The amount of memory is never used to hold more than \(O(|S|)\) strands, each of length \(n\).

## 5 Shuffle Codes with Constant GC Contents

It is desirable in practice to impose an additional requirement that DNA strands have roughly the same amount of G’s and C’s [3], due to the fact the number of GC bonds (which are stronger than AT bonds) is a good measure of melting characteristics of DNA strands. Thus, DNA strands with similar amounts of GC’s have similar melting characteristics and would serve better biomolecular experiments that rely heavily on repeated melting and annealing DNA strands, such as PCR selection. Using the less realistic DNA-Hamming codes, King [19] established upper and lower bounds for code sizes with constant GC content. These results do not immediately carry over to the space defined by our \(h\)-
measure, except in the case when \(\tau = 1\). Define an \((n, \tau, w)\)-code to be an 
\((n, \tau)\)-code with an additional requirement that every strand must have exactly \(w\) C’s and \(G\)’s in total.

**Theorem 5.** The shuffling construction specified in Algorithm 2 can be used to 
construct an \((\tau m, \tau, \tau)\)-code of size \(m \tau 2^{(m-1)\tau}\). Generally, Algorithm 2 can be
Algorithm 2 Input: $\tau,m,w$. Output: an $(\tau m,\tau tw)$-code

Construct $S_m$, the set of all $m$-strands having exactly $w$ G’s and C’s in total.

\{note: $|S_m| = \binom{m}{w} 2^m$\}

for all strand $x \in S_{m,w}$ do

Construct $x^\tau$, the unique complementary of $x$, i.e. $h(x, x^\tau) = 0$.

$S_{m,x} = S_{m,w} - x^\tau$

end for 

(note: $|S_m| = \binom{m}{w} 2^{m-|Q_m|}$)

Let $S = \{x_1 \otimes x_2 \otimes \cdots \otimes x_r : \forall x_i \in S_{m,w} \text{ and } x_i \neq x_j \text{ if } i \neq j\}$

return $S$.

used to construct an $(\tau m,\tau tw)$-code of size

$$\left(\frac{\binom{m}{w} 2^m - |Q_m|}{2}\right)^\tau$$

where $|Q_m| = \binom{m/2}{w/2} 2^{m/2}$ if $n$ and $w$ are even and $|Q_m| = 0$ if $m$ or $w$ is odd.

Proof. The set $S_m$ constructed by Algorithm 2 is an $(m,1,w)$-code of size $\frac{\binom{m}{w} 2^m - |Q_m|}{2}$, where $Q_m = \{x : h(x, x) = 0, x \text{ has exactly } w \text{ G’s or C’s in total}\}$, and $|Q_m| = \binom{m/2}{w/2} 2^{m/2}$ if $m$ and $w$ are even, and 0 if either $m$ or $w$ is odd. This is because there are $\binom{m}{w}$ configurations for G’s and C’s, in each of which there are exactly two choices for each position. Then, we need to discount the strands of measure 0. As before, these include the Watson-Crick palindromes $(Q_m)$ and the unique complementary strands of all the strands left. Further, the number of Watson-Crick palindromes is exactly $\binom{m/2}{w/2} 2^{m/2}$. This is because the $i^{th}$ and $(m - i + 1)^{th}$ characters of these strands must be complementary, which implies $m$ and $w$ must be both even and further the GC-content of $w$ implies that each position has exactly 2 choices. Finally, as in the proof of theorem 2, the shuffling construction scales linearly both stringency and GC-content. In other words, a shuffle of $\tau (m,1,w)$-codes yields an $(\tau m,\tau tw)$-code. \hfill \Box

Again, the shuffling construction yields a close-to-optimal code when $\tau$ is small, as proved above:

Theorem 6. Let $L_\tau$ be a maximal $(\tau m,\tau tw)$-code, and $S$ be the $(\tau m,\tau tw)$-code constructed as above. If $\tau m$ or $\tau tw$ is odd, then

$$2^{\tau - 1} \cdot |S| \geq |L_{\tau}|$$

Proof. $L_1$, the maximal $(\tau m,1,\tau w)$-code, is larger than $L_\tau$, and thus $\frac{|S|}{|L_1|} \leq \frac{|L_{\tau}|}{|L_1|}$. Using Sterling’s approximation to factorials ($\binom{m}{w} \approx \binom{m}{w} \frac{m!}{w!(m-w)!}$) we get:

$$\frac{\binom{m}{w}}{2^{\tau - 1} \binom{\tau m}{w}} \leq \frac{(\binom{m}{w} 2^m)^{\tau}}{2^{\tau - 1} \binom{\tau m}{w} 2^{\tau m}} = \frac{|S|}{|L_{\tau}|} \leq \frac{|L_{\tau}|}{|L_1|}$$

\hfill \Box
Theorem 7. Algorithm 2 takes \(O(n \cdot |S|)\) steps and uses \(O(n \cdot |S|)\) memory, where \(S\) is the \((\tau m, \tau, \tau w)\)-code returned by the algorithm.

Proof. Similar to the proof of Theorem 4.

6 Capacity of DNA-based Associative Memory

Garzon et al [13] propose to use DNA arrays as a medium to store abiotic information. In this model of associative DNA memory model, stored information is hybridized to on-the-chip sequences. The memory of each input is defined as the set of subsequences hybridized by the on-the-chip sequences. Therefore, the more mutually orthogonal these on-the-chip sequences are, the less overlap they hybridize to memory inputs, and potentially more inputs are distinguishable. Using Algorithm 1 or 2, we can construct such large codes.

We now derive an information theoretic bound on the probability of two inputs being indistinguishable. Let \(S = \{s_1, s_2, \ldots, s_k\}\) be the on-the-chip sequences. Given an input \(X\) to be stored in memory, define \(X_S\) to be the image of \(X\) on \(S\) as \(X_S = (x_1, x_2, \ldots, x_k)\), where \(x_i\) is the number of times \(s_i\) hybridizing to (different parts) of \(X\). Inputs \(X\) and \(Y\) are indistinguishable in \(S\) if \(X_S = Y_S\), i.e. \(x_i = y_i, 1 \leq i \leq k\).

\[
P(X_S = Y_S | X \neq Y) = \left( \frac{x_1, x_2, \ldots, x_k}{k^\sigma} \right) \leq \frac{2^\sigma H(P)}{k^\sigma} = \frac{1}{2^\sigma (\log_2 k - H(P))}
\]

where \(\sum_{i=1}^{k} x_i = \sum_{i=1}^{k} y_i = \sigma\), and \(H(P) = -\sum_{i=1}^{k} \frac{x_i}{\sigma} \log_2 \frac{x_i}{\sigma}\), the Shannon entropy of the distribution of \(s_i\) in \(X\) (and \(Y\)). This implies that if the distribution of the code \((s_i's)\) in the input is random (i.e. \(H(P) \rightarrow \log_2 k\)), then using \(S\) as a memory is not effective. Therefore, we have two objectives in designing on-the-chip sequences in an associative DNA memory model: (1) make them code and as large as possible, and (2) their distribution as subsequences of the stored information is as far from uniformity as possible.

References