The University of Memphis   
Computer Science   
COMP 7612 FOUNDATIONS OF COMPUTING  
Project Proposal:  **DNA CODEWORD DESIGN  
Senjuti Dutta, Computer Science, Spring 2018**

**GOAL AND SCOPE**   
The goal of the project is to find an approximate solution of CODEWORD DESIGN problem using reduction as well as provide a proof of correctness of reduction and complexity analysis of the problem.

**DNA CODEWORD DESIGN**  
 INSTANCE: Two integers n / 𝛕 (size of the probes/hybridization stringency)  
 QUESTION: Find a selection of sphere centers (probes) in the DNA space such that every other string of string size n is within *h*-distance 𝛕 of exactly one probe.

**LIT REVIEW**

In 1953 Watson and Crick discovered the role and structure of DNA. This discovery ushered in a new era for the biological sciences that continues unabated despite enormous progress for well over half a century now. The success of the human genome project of the 1990s marked a transformation in the theory and practice of experimental biology. A fundamental property of DNA is hybridization, which depends on Gibbs energies of duplex formation. The h-distance is a model of Gibbs energy with metric properties. The main sources of information about codeword design and codeword designing has been researched before the design of this codeword is undertaken, beginning with [1,2,3]. These and other sources has been used as needed to complete this project successfully.

**DNA Codeword Design Requirement:**

1. Success in specific hybridization between a DNA codeword and its complement.
2. No hybridization between DNA codewords from same DNA or between a DNA codeword and others’ Watson-Crick complements.
3. Important in DNA Computing, self-assembly, DNA memories and phylogenetic analyses.

**DNA CODEWORD Design Application:**

In case of DNA computers, the basic operation of the DNA molecules is hybridization and hybridization it is very important to select DNA molecules so that they can hybridize and can form a double helix structure. DNA CODEWORD DESIGN problem is very helpful to select molecules as probe to build more powerful DNA computers.

To understand the concept of Codeword design, concepts of Gibbs energy, h-distance need to be explained first.

**Gibbs Energy:**

A fundamental property of DNA is hybridization, which depends on Gibbs energies of duplex formation. Here how the Gibbs energy is related to duplex formation: [1]

The energy released or absorbed in the hybridization of two DNA strands is related with the familiar Gibbs energy in thermodynamics and physics, generally given by G (T, p) = U + pV - TS, that depends on the internal energy (U), pressure (p), volume (V), temperature (T), and entropy (S) of the environment in which the duplex is formed. The more negative the Gibbs energy, the more stable the duplex formed.

According to the nearest-neighbor (NN) model the free energy for duplex formation basically depends on three factors: the initial free energy given by the unfavorable entropy due to the loss of translational freedom, the sum of the complementary pair-wise terms between the nucleotides sequences being stacked to form the duplex, and the entropic penalty of the maintenance of the symmetry in a double strand which is a factor of hybridization affinity.

Gibbs energy does not follow the triangle inequality ac<= ab + bc (where a, b, c are three points in a triangle). In triangle the distance from point a to point b and distance from point b to point a should be same whereas in case of Gibbs energy the energy required to go from point a to point b and point b and point are not same. Therefore, a different approach via the h-distance which provides a computationally efficient approximation of the Gibbs energy based solely on composition and sequence is introduced as a more theoretical and approximate concept.

**h-distance Model**

In this model the appropriate chemistry is abstracted as two operations as follows: [2]

1. unary operation between Watson-Crick complementation
2. a binary operation of hybridization.

The Watson-Crick complement y’ of strand y is obtained by reversing it and swapping nucleotides within the pairs a-t and c-g. Hybridization is modeled as a binary operation between two single strands x, y that forms a double-stranded molecule in which pairs of nucleotides are joined in a double helix form by hydrogen bonds, for an a = t or a c = g pair, respectively.

The distance is defined in terms of the h-measure between two single strands x, y as follows:

(1)

where σ k is the shift of y by k positions (right-shift if k > 0; left-shift if k < 0) with respect to x, y' is the Watson-Crick complement of y. The Hamming distance H measures the number of different bases (non-complementary in the original pair) in the overlap of x and y' in the specified frameshift σ k (y'). The h-distance is a number assigned between two DNA strands which describes the likelihood of hybridization between two strands. For example, when h-distance is 0 then the DNA strands are perfectly complement to each other and there is maximum likelihood of hybridization between them. The smaller the h-distance, the more likely are the strands to hybridize.

**Example:**

A measure of h (x, y) = 0 means x = y’. A large distance indicates that even when x finds itself in the proximity of y, they contain few complementary base pairs regardless of the position they find themselves in, and thus are unlikely to form a stable duplex, i.e. no stable hybridization is likely to take hold. For example, if x = agc, y = tgg (and y’ = cca), at shift k = -2

, there are 2 pair mismatches and one identity match 1 + H (a, a) = 2; at shift k = -1

there are 3 pair mismatches 1 + H (ag, ca) = 3; at shift k =0 (in perfect alignment

there are 3 mismatches H (agc, cca) = 3; at shift k = 1 the number of mismatches is

1 + H (gc, cc) = 2; and at shift k = 2, the distance is 2 + H (c, c) = 2. Thus h (agc, ttg) = 2.

**CASE STUDY**

CODEWORD DESIGN is reducible to a sphere packing problem in Euclidean space. The reduction procedure is described below: [3]

First stage in the reduction is a representation of DNA strands x **D**n as vectors in the Euclidean space **R**4n of dimension 4n which implies:

n(an instance of DNA CODEWORD DESIGN) -> 4n (n -D space in EUCLIDEAN SPHERE PACKING) (2)

Given a basis b {a, c, g, t}. Let Ψb(x) **R**4n be the vector given by Ψb = Ψ1 if b = xi otherwise 0. Ψb indicatesthe presence or absence of nucleotide b at position i in x. A DNA oligonucleotide x is represented by Ψ and DNA oligonucleotide y is represented by Ψ in Euclidean space.

The representation Ψ of a strand x is given by the concatenation of these vectors in the following order

Ψ(x)

For example, 1000, 0100, = 0010, (x) = 0001 and x = tgca is represented as 0001001001001000.

Every pair of strands x, y where ||Ψ|| is the standard Euclidean norm of a vector, “." Denotes the standard dot (scalar) product between two vectors in **R**4n, and |x| denotes the length of strand x.

Strands of length n are represented as points on a sphere of radius that means

| |Ψ2 = |x| = n and || Ψ|| = (3)

The number of Watson-Crick complementary matches between x and y in perfect alignment is m (x, y) = Ψ.

H (x, y) = n - Ψ. (4)

The Hamming distance can be normalized as

Ψ

= Ψ) = 2 sin 2 (5)

where is the angle between the two 4nD vectors Ψ . This metric is essentially equivalent to H and has the advantage that it represents DNA strands as Euclidean points on the unit sphere in the Euclidean space of dimension 4n.

When DNA-strands are perfectly aligned with each other the h-distance becomes -distance. It is then computed by just the simple normalized Hamming distance where the two vectors x, y lie on the same side of the mirror (the invariant subspace), i.e.,

(x, y) = 2 sin2() (7)

From equation (1) and (7) we can get the concept of following equation:

𝛕 ≤ 2 sin2() (8)

As a result, maximal DNA codeword sets of n-mers of noncrosshybridizing quality given by a minimum separating h0- distance of 𝛕 is reduced to Euclidean sphere packing problem.

**TAKE-HOME DELIVERABLE**

**Proof of Correctness of Reduction:**

The reduction is an algorithm where one instance of one algorithm is transformed to an instance of another algorithm and is correct if answers to the decision questions are always “YES” for positive instances and “NO” for negative instances.

DNA CODEWORD DESIGN (Decision Version)

INSTANCE: A set of n-mers, a threshold 𝛕 and an integer n

QUESTION: Is there (n, 𝛕) code of k-elements or less?

EUCLIDEAN SPHERE PACKING (Decision Version)

INSTANCE: A Euclidean dimension n, α, angle

QUESTION: In n-D Euclidean space is there any arrangement of spheres of angle so that it covers α fraction of the whole hypercube?

Instance in DNA CODEWORD DESIGN is (n, 𝛕) and instance in EUCLIDEAN SPHERE PACKING is (n,.The mapping of instance of CODEWORD DESIGN to EUCLIDEAN SPHERE PACKING is already shown in equation (2) and (8).

So, if there exists a code (n, 𝛕) in DNA space there also exists (n,in Euclidean space. So, the answers for the positive instance to both the decision questions of CODEWORD DESIGN and EUCLIDEAN SPHERE packing will be same and it is “YES” as an example f n = 4, 𝛕 = 2 in DNA space then n = 16 and = π in Euclidean space. It is clearly understood that if there does not exist a code (n, 𝛕) in DNA space then there does not exist any (n, for Euclidean space and the answers for the negative instance to both the decision questions are same “NO”. For example, if n =-50, 𝛕 = 0 in DNA space then n = -200 = complex number in Euclidean space. Thus, the reduction is correct.

**Approximate solution of CODEWORD DESIGN:**

Reduction of CODEWORD DESIGN to EUCLIDEAN SPHERE PACKING indicates solution to EUCLIDEAN SPHERE PACKING for particular n and is a solution to DNA CODEWORD DESIGN for corresponding (n, 𝛕). There are solutions for Euclidean dimension n like as n =1, n=2, n=3 (the problem is famous for sphere kissing problem of Newton), n= 24 (the problem is famous for unique solution of 196,540 kissing 24-D spheres).

According to the paper [2], the best-known solutions are afforded by so-called lattice packings, obtained by diophantine (integer coefficients) combinations of n linearly independent vectors (usually assumed to be diophantine as well). As an example, to obtain solution of CODEWORD DESIGN with a DNA codeword set from a known optimal solution for lattice packings in 24D-space (i.e., for 6-mers), the Leech Lattice generated from Golay’s [4,5] 3-error correcting code t in 24D Hamming space and is known to have a separation angle of arccos (1/3) ≈70° there are used the following steps:

* Decompose the lattice points in 4D-space into nearly concentric layers(shells) around one of its points located as origin
* The points in this shell are then projected onto **Ω**4n by dividing by their norm. They provide an approximate solution to the sphere kissing problem.
* DNA codeword sets to be guaranteed to be at a minimum angular distance 𝛕 will be obtained by filtering from this kissing set 4n-dimensional points coding for DNA strands that are within a suitable angle (usually k = 2 or 3 will do) of a (normalized) shell point, for a suitable value of k Here = | – Arccos (1/3) | is the difference between the angle separation of two lattice points and the desired angle separation between DNA codewords.

Thus, well established solutions to spherical coding problems can be translated into codeword design solutions that are in some cases known to be nearly optimal

**Time Complexity:**

To solve the complexity of the problem here decision version of the CODEWORD DESIGN problem is taken into account which is already explained above.

To prove that this problem is in NP there needs to be sown how the Turing Machine (TM) works.

The TM M works as follows:

1. Nondeterministically guesses no of probes say m
2. Nondeterministically guesses m DNA strings of length n for potential (n, 𝛕)
3. Compute h-distance between any pairs of those probes and checks if the h-distance is at least 𝛕
4. Checks the no of probes is less than k or not

When the machine computes the h-distance between first pair of those m probes and if the h-distance is at least 𝛕 then the machine goes for next pair and computes the h-distance and if it is also at least 𝛕 and the machine will continue and if the condition for h-distance at least 𝛕 satisfies for all of the pairs then the machine will clear the work tape, print “YES” and halt. The machine will also check that the number of probes is less than k or not if it is less than k then the machine will clear the work tape, print “YES” and halt.

If there does not exist such no of probes where the h-distance between every pair of probe is not at least 𝛕 then the machine will print nothing, clear the work tape and halt and if the no of probes is not less than k then the machine will print nothing, clear the work tape and halt. It means there does not exist such (n, 𝛕) code. Here Resource = Number of calls to compute h-distance, input size, n = n (size of the string) so time complexity, tn = O (n2k2) which is nondeterministic polynomial time. So, time complexity of this problem is in NP.

According to the paper [1] to prove that the CODEWORD design problem is NP-hard this decision problem is reduced from 3-SAT problem To prove NP-hard there is a transformation of in polynomial time to a graph and need to show that is satisfiable if and only if has a clique of size .It is given ,F(xi) = and F = where is a (log mer consisting of only a’s and g’s and is the reverse complement of .F(Ci) = Ti is a set of three (logk)-mers, Ci = C1 …, Cm. The union of these Ti ‘s form the vertices of the graph . An edge between si and sj if and only if vertices belong to different Ti’s and the corresponding variables xi and xj are not negations of each other, so that d (, ) > 0. Here resource = time to construct edges, input size, n = k (no of elements in (n, 𝛕) code) so = O(k2). It is claimed that is satisfiable if and only if has a clique of size k, i.e. is an (n,1)-code. If is satisfiable, then each clause Ci has at least one literal that is true. There are the following steps to show that is satisfied:

1. Create a set Q by selecting exactly one strand sj from each Ti such that sj corresponds to the variable xj that is true in Ci.
2. Q is a k-clique because any two vertices in Q are not Watson-Crick complements of each other as their corresponding literals are both true and consequently cannot be the negation of each other.

Conversely, if has a clique of size k, then the k vertices must belong to different groups, namely k of the Ti’s. By assigning the corresponding literals to be true, the formula is satisfied. This proves the decision problem of Codeword Design is in NP hard. CODEWORD DESIGN problem is in NP and at least as hard as SAT problem. So, this problem is NP-complete.

**Deterministic Time Complexity:**

To get deterministic time complexity of the decision version of CODEWORD DESIGN problem I will mention that for 60 years many researchers are trying to convert nondeterministic polynomial time solution to deterministic polynomial time solution but except primality testing the researchers are not able to convert any nondeterministic polynomial time problem to deterministic polynomial time problem. In this situation there are three types of scenario:

1. There may be exist a solution but researchers are not able to find it
2. There does not exist a solution
3. We do not know if there exist a solution or not

In this case the deterministic complexity is - we do not know if there exists a solution or not.

**REFERENCES**

[1] Garzon, M., and V. Phan. "On codeword design in metric spaces." Natural Computing 8.3 (2009): 571-588

[2] Garzon, Max H. "DNA codeword design: Theory and applications." Parallel Processing Letters 24.02 (2014): 1440001.

[3] Garzon, Max H., and Kiran C. Bobba. "A geometric approach to Gibbs energy landscapes and optimal DNA codeword design." International Workshop on DNA-Based Computers. Springer, Berlin, Heidelberg, 2012.

[4] M.H. Garzon, K.C. Bobba, A.J. Neel and V. Phan, DNA-Based Indexing, Nanotechnology and Molecular Computation 2:3 (2010) 25-45.

[5] V. Phan and M.H. Garzon, On Codeword Design in Metric DNA Spaces, J. Natural

Computing, 8:3 (2009) 571 - 588.