Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

On DNA Computers Controlling Gene Expression Levels

Olgica Milenkovic
Dept. Electrical and Computer Eng.
University of Colorado
Boulder, CO, 80309, USA.

Navin Kashyap
Dept. Mathematics and Statistics
Queen's University
Kingston, ON K7L3N6, Canada.

Bane Vasic
Dept. Electrical and Computer Eng.
University of Arizona
Tucson, AZ 85721, USA.

Abstract— **The area of bio-molecular computing has recently witnessed a major paradigm shift. Rather then being used only as simple calculating units capable of solving hard combinatorial or numerical problems, DNA computers are increasingly becoming more tailored to operate like intelligent biological machines with unprecedented potentials. One example of applying DNA computers in such a new setting is in the area of logical control of gene expression levels. For this purpose, DNA computers are designed in such a way as to be able to diagnose some forms of cancer-related irregularities in a cell and release biological strands acting as inhibitors or activators of certain sets of genes. Such a control action can also be seen as a form of intra-cell cancer therapy, although it may also have other, more varied, purposes and goals. There are several important problems in the area of coding and network theory that arise in the context of developing DNA computers for controlling gene expressions. The two most important issues are that of minimizing diagnostics failure and of increasing the computational reliability of the system. The first question is intimately related to analyzing the operational principles of networks of gene interactions, while the second is concerned with relating combinatorial characteristics of single-stranded DNA sequences to their hybridization affinities and secondary structures. In this paper, we will describe the state-of-the-art results and present some new relevant combinatorial and coding theoretic problems in this area.**

I. Introduction

The last century was marked by the birth of two major scientific and engineering disciplines: silicon-based computing and the theory and technology of genetic data analysis. The research field very likely to dominate the area of scientific computing in the foreseeable future is the merger of these two disciplines, leading to unprecedented possibilities for applications in diverse areas of biological and medical sciences. The first steps toward this goal were made in 1994, when Leonard Adleman [1] experimentally demonstrated the computational potential of biological macromolecules. DNA computing introduced the possibility of using genetic data to tackle computationally hard classes of problems that are impossible to solve efficiently using traditional computing methods. The biological properties that make DNA computers capable of achieving these goals are massive parallelism of hybridization and enzyme-based operations on nano-scale, low-power molecular hardware and software systems. More importantly, practical demonstrations of DNA computing principles opened new venues for applications of bio-devices for disease diagnostics and treatment. A landmark development in this area was the construction of an autonomous DNA machine for controlling the expression levels of genes [2]. Control of gene expression profiles appears in every biological system, either at the level of the DNA sequence (in terms of nucleotide rearrangements), the level of transcription or at the level of post-transcription [6]. The key idea of such a control process is that a gene is not active as long it is not transcribed or as long as the result of its transcription is rendered un-functional. Post-transcriptional silencing of genes, in terms of RNA Interference (RNAi) is currently one of the most outstanding research topic in molecular biology [24].

The principles underlying post-transcriptional expression regulation were implemented in the DNA computer system described in [2]. The DNA automata used for this purpose was designed to act like an *RNA fingerprinting device*. Here, fingerprinting refers to a set of processes aimed at detecting the presence and concentration of specific mRNA molecules in the cell. These RNA molecules carry the information for protein encoding and therefore serve as indicators of the activity of their corresponding genes. If, for example, mRNA sequences carrying the messages of genes PPAP2B and GSTP1 are under-represented, while mRNA sequences carrying the messages of genes PIM1 and HPN are over-represented within a cell in the prostate, there exists a high probability that the cell is undergoing cancerous changes. In this case, the DNA computer is instructed to administer a short DNA sequence – GTTGGTATTGGACATG – that inhibits the generation of a protein (MDM2) and therefore changes the interaction patterns between the genes involved [2].

Several coding theoretic issues are apparent when investigating the operation of DNA computers designed to control gene expressions. One aspect is connected to the functionality of the *disease identifier* block. This part of the system architecture is in charge of estimating specific

0-7803-9568-9/05/$20.00 ©2005 IEEE

levels of a subset of mRNA molecules in a cell. In order for this system to work reliably, one has to have a sufficiently accurate set of disease indicators (i.e. a sufficiently large and discriminatory list of genes that are under- or over-expressed in the cell under investigation), and a reliable estimate of what expression levels are to be considered "low" and "high". The situation is further complicated by the fact that expression levels in affected cells usually vary in time and are genuine random variables, either due to the influence of external factors or due to changes associated with normal cell cycle processes. Another aspect is connected to the fact that single-stranded DNA sequence operations are prone to various types of hybridization and self-hybridization errors and instabilities. DNA computers have to operate in a controlled environment that allows for a set of single-stranded DNA codewords to bind (hybridize) only with their exact complements. Furthermore, for certain applications, DNA strands may be required to avoid secondary structures - i.e. folding back onto themselves. If such environment cannot be established, as may be the case for *in vivo* computations, unwanted, or non-selective, hybridization may occur. Consequently, one has to carefully design DNA sequences of appropriate length and structure in order to ensure optimal operational conditions.

Undoubtedly, all the aforementioned problems have to be first addressed at the level of (experimental) molecular biology in order to find all necessary data needed for devising the computer's architecture. But it is also very important to develop a mathematical approach for modelling genetic data and representing it in a way that can be used within DNA computers. The first task, concerned with analyzing and modelling gene regulatory networks in terms of random Boolean networks, neural networks and/or stochastic automata was addressed in a series of papers [9], [11], [19], [20] and the references therein. On the other hand, the focus of error-control coding research for DNA computing was so far directed mainly towards constructing *large* sets of DNA codewords that have a small probability of undesired hybridization; Two design principles were put forward. The first principle is based on the assumption that the backbone of DNA strands is a perfectly rigid structure. Such an assumption implies that fixed base frequency (constant GC-content) and prescribed Hamming/reverse-complement Hamming distance constraints can minimize hybridization-related problems [8], [12]. The second approach is based on the assumption that the backbone of DNA molecules is perfectly elastic. In this setting, unwanted hybridization can be minimized by using DNA codewords designed according to principles that govern constructions of classical deletion-correcting codes [7]. Under the two described models, DNA codewords obeying the given constraints are expected to very rarely hybridize in an erroneous fashion. Unfortunately, such models may fail to predict the correct hybridization pattern due to the fact that the sugar-phosphate backbone of DNA strands has physical and thermodynamical properties that are not adequately captured by a completely rigid or elastic structural model, and due to the fact that the stability of a DNA duplex depends on the nearest neighbor interaction energies described in [3]. Additionally, few attempts were made to test codewords designed with respect to the hybridization constraint for secondary structure formations, nor was the folding constraint incorporated in any of the proposed code construction methods directly. For the problem at hand, the importance of the hybridization and secondary structure constraint is even more apparent due to the fact that time-varying folding properties of single-stranded and double-stranded DNA or RNA molecules are important pointers towards their time-changing function [22].

The two classes of problems described above have been addressed using diverse approaches from statistical physics [4], [16], computer science [18], biology and mathematics. Gene regulatory systems were modelled in terms of random Boolean networks, neural networks, differential equations and stochastic equations [11]; DNA hybridization and folding dynamics were analyzed computationally, by using specialized modifications of dynamic programming algorithms [3]. Here, we will focus our interest on the *combinatorial* and *coding theoretic aspects* of such problems. More specifically, we will describe techniques for modelling gene regulatory networks and quantifying DNA hybridization and folding properties based on ideas borrowed from coding theory. Our treatment of the two problems is based on viewing regulatory networks as iterative systems and analyzing the combinatorial aspects of DNA folding in terms of the nearest neighbor results of [3] and some characteristic properties of Nussinov's algorithm [17].

The paper is organized as follows. Section II contains a description of relevant concepts from the theory of gene regulatory networks (GRNs) and and overview of prior work on modelling such systems. The same section also describes some alternative analytical techniques for analyzing GRNs inspired by results from the theory of iterative decoding [5]. Section III provides the necessary background needed for analyzing the combinatorial aspects of DNA hybridization and folding and a sampling of results regarding DNA codeword constructions.

II. Gene Regulatory Networks: Definitions and Terminology

Although DNA molecules can be simplistically viewed as (one-dimensional) sequences over a four-letter alphabet, there exists many higher levels of interaction within such strings. Consequently, DNA sequences should be viewed as networks of interacting subunits. In this setting, it is often assumed that the main functional subunits are genes. Genes are subsequences of DNA strands that code for proteins that are essential for the development and functioning of an organism. The central dogma of genetics represents a set of rules according to which genes are transcribed into mRNA molecules, subsequently used for encoding of proteins during the process of translation. The expression level of a gene is a measure of the frequency with which it is being transcribed, and represents a crucial parameter for quantifying its activity. Genes in regulatory networks interact indirectly through enzymes and other regulatory complexes, which are very often themselves encoded by genes. There exist many techniques for modelling the interaction patterns of genes in so-called *gene regulatory networks* (GRN). Among the most frequently used methods are differential equations for describing positive and negative feedback actions [11], and random Boolean and neural networks [19], [20]. In the former case, the process of regulation is described in terms of kinetic equations of the form:

$$\dot{s}_i = f_i(s_1, ..., s_N),\ i = 1, ..., N,$$

where f_i is the expression-level law (equivalently, rate-law), N denotes the number of genes $G_i, i = 1, ..., N$, and $s_1, ..., s_N$ describe the concentrations of the gene products. Although in such a formal set-up the equations cannot be solved directly, there exist many numerical methods that can be used for obtaining qualitative results for gene activities. In the latter case, one defines a regulatory message $r_i(t)$ for a gene G_i at time t as [20]:

$$r_i(t) = \sum_{j=1}^{N} \omega_{i,j} s_i(t) + e_i,\ i = 1, ..., N,$$

where $\omega_{i,j}$ denotes the strength of the influence of gene G_j on gene G_i, while e_i described the strength of the influence of external factors on G_i. If gene G_j exhibits no influence on G_i, then $\omega_{i,j} = 0$. Otherwise, $\omega_{i,j}$ is either positive or negative, according to the regulation being responsible for increasing or decreasing the expression level of gene G_i. The transcriptional response of G_i is modelled in terms of the sigmoidal transfer function [20] $g_i(t) = 1/(1 + e^{-r_i(t)})$, while the expression level of the gene is of the form $s_i\, g_i(t)$, where s_i stands for the maximum achievable expression rate of G_i. Some drawbacks of the described models are that they do not capture the fact that interactions between genes occur at random time intervals, and that the expression levels of the genes themselves represent random variables [23]. For example, the expression levels of a gene within two different cells of the same type, and in the same cell-cycle stage, can be quite different. This can be attributed to the probabilistic nature of biochemical reactions as well as to random external conditions. This makes the regulatory process stochastic in nature. Stochastic differential equation models for gene regulatory networks were considered in [21]. There, stochastic equations were used to describe the time evolution of a genetic systems in terms of the updates of the state variable, $S(T) = [S_1(T), ..., S_N(T)]$, representing the stochastic expression levels of the genes at time T. Let $P_j(\Delta, S)$ denote the probability of the j-th regulatory reaction occurring in the time interval $[T, T + \Delta T]$, provided that the network is in state S. Similarly, let $P_j(\Delta, S)$ denote the probability that the terminal state of the network after completion of reaction j will be S. Based on a set of plausible reactions, one can obtain computationally the sequence of states through which the genetic system goes through according to the updating rule:

$$P\{S(T + \Delta T)\} = P\{S(T)\} \left(1 - \sum_{i=1}^{m} P_j(\Delta, S)\right) + \sum_{i=1}^{m} Q_j(\Delta, S).$$

The drawbacks of the differential equation models are mainly connected to the their large computational complexity, which itself is a consequence of the analytic intractability of the model. This is why one of the most commonly used models for gene regulatory interactions is based on random Boolean networks or randomly perturbed neural networks. There exist well-developed techniques for the analysis of Boolean networks that are largely taken from the theory of discrete dynamical systems. In the next section we will describe how the theory of message passing borrowed from coding theory can be used to analytically approach the problem of neural network modelling. Furthermore, we will describe how *density evolution* [5] can be used to quantify iterative systems for time-dependent average gene expression levels.

A. Gene Regulatory Networks as Iterative Message Passing Systems

We propose to analyze GRNs in terms of a Gaussian neural network model. A model of this type was first put forward in [20]. Here, we will add some new and biologically relevant features into the model [23], and determine its properties based on techniques borrowed from the theory of iterative decoding. The underlying connection between the biological and coding-theoretic entities is the inherently dynamic (iterative) change of state variables in the system. For a coding application, such changes are aimed at determining the correct message values. For biological systems, they are performed in a manner that allows cell processes to proceed in a correct manner. The key idea behind the analysis is based on *density evolution* (DE) with Gaussian approximation [5]. In a DE-type of analysis, one assumes that at each time instant, the expression level is a Gaussian variable. Under this assumption, one can track only the time changes of the first and second moment of the variables. Accordingly, the time changing and asymptotic values of the parameters of the variables can be easily determined. Even in the case when the variables involved are not Gaussian, the moment-tracking method may still provide good predictions of the system's dynamics. Consider the following model of gene expression level updates (which is a combination of the models in [20] and [23]):

$$\begin{aligned} &u_{i,n+1} = a_{i,n} \cdot u_{i,n} + g_{i,n} \cdot b_{i,n}, \\ &a_{i,n} \sim \mathcal{N}\left(\mu_{i,n}, M_{i,n}^2\right), \;\; b_{i,n} \sim \mathcal{N}\left(\beta_{i,n}, \Sigma_{i,n}^2\right), \\ &g_{i,n} = 1/\left(1 + e^{-\sum \, \omega_{i,j} u_{j,n-\tau(j)} + \alpha_{i,n}}\right), \\ &\alpha_{i,n} \sim \mathcal{N}\left(0, \Gamma_{i,n}^2\right), \;\; \omega_{i,i} = 0 \;\; \forall\, i. \end{aligned} \tag{1}$$

In the previous equation, the random variables $a_{i,n}$, $b_{i,n}$ and $\alpha_{i,n}$ are assumed to be independent Gaussian variables that are also independent from the variables $u_{i,n}$. They implicitly contain the information about the degradation rate of a gene's mRNA and about the maximal expression rate of a gene. The parameters $\omega_{i,j}$ are to be determined experimentally, for example by using DNA microarray measurements. The correction term $\alpha_{i,n}$ accounts for external factors that influence the expression levels of genes as well as possible stochastic changes in the weighting factors $\omega_{i,j}$. Furthermore, it is assumed that a gene cannot exhibit direct influence on itself, so that $\omega_{i,i} = 0$. Finally, the integer time shifts $\tau(j)$ are introduced in order to allow for different "propagation times" of biochemical reactions initiated by gene G_j and influencing gene G_i. For simplicity, these parameters will be taken to be independent from the time (iteration) variable n. The formula in (1) is reminiscent of the belief propagation equation of iterative decoding. The major difference between the two expressions comes from the form of the function $g_{i,n}$, which in the coding theoretic setting is significantly more complicated. The expected value $m_{i,n}$ and second moment $\theta_{i,n}$ of the expression levels $u_{i,n}$ can be found as

$$\begin{aligned} &m_{i,n+1} = \mu_{i,n}\, m_{i,n} + \beta_{i,n}\, E[g_{i,n}], \\ &\theta_{i,n+1} = \left(M_{i,n}^2 + \mu_{i,n}^2\right)\, \theta_{i,n} + \left(\Sigma_{i,n}^2 + \beta_{i,n}^2\right)\, E[g_{i,n}^2]. \end{aligned}$$

Using similar arguments as in the density-evolution approach of [5], one can obtain approximations for the average expression levels in the following manner. First, simple upper and lower bounds for the integrals involved are computed, which are then used to find an approximation in terms of their weighted average. To find such tight bounds, one can utilize the following representation of the sigmoid function [10]

$$\frac{1}{1+e^{-x}} = \min_{\zeta \in [0,1]} \; e^{\zeta\, x - H(\zeta)},$$

where $H(\zeta) = -\zeta \, \log \, \zeta - (1-\zeta) \, \log\,(1-\zeta)$ is the binary entropy function. As a result, one has

$$\frac{1}{1+e^{-x}} \leq e^{\zeta\, x - H(\zeta)},$$

and

$$\frac{1}{1+e^{-x}} \geq \frac{1}{1+e^{-\eta}}\, e^{(x-\eta)/2 - F(\eta)(x^2-\eta^2)},$$

where ζ is any number in $[0, 1]$, η is arbitrary and

$$F(\eta) = \left(\frac{1}{1+e^{-\eta}} - 1/2\right)/2\eta.$$

This implies that

$$\begin{aligned} &E[g_{i,n}] \simeq \frac{1}{2}\exp(U_i\zeta + \zeta^2 V_i^2/2 - H(\zeta)) + \\ &\frac{1}{2}\exp\left(\frac{(U_i + V_i/2)^2 - U_i^2(1+2V_i^2\, F(\eta))^2}{2V_i^2(1+2V_i^2\, F(\eta))}\right) \times \\ &\frac{\exp(\eta/2 + F(\eta)\eta^2/2)}{1+e^{-\eta}}, \\ &U_i = \sum_{j=1}^{N} \omega_{i,j}\, m_{j,\tau(j)}, \\ &V_i^2 = \Gamma_{i,n}^2 + \sum_{j=1}^{N} \omega_{i,j}^2\, \left(\theta_{j,\tau(j)} - m_{j,\tau(j)}^2\right); \end{aligned}$$

The parameters ζ and θ are to be chosen in such a way as to minimize the approximation error. Note that similar expressions can be derived for $E[g_{i,n}^2]$. These two approximations provides for a simple iterative system for which convergence rates, activation levels and other important features can be determined by using the density evolution approach [5].

III. DNA Coding: Definitions and Terminology

DNA of higher species consists of two complementary chains twisted around each other in the form a double helix. Each chain consists of a *backbone*, composed of sugar and phosphate units, and a linear sequence of nucleotides, or *bases*. Two of the bases are of the *purine*-type, namely adenine (**A**) and guanine (**G**), while the other two are of the *pyrimidine*-type, namely thymine (**T**) and cytosine (**C**). The purine bases and pyrimindine bases are *Watson-Crick (WC) complements* of each other, in the sense that

$$\overline{\mathbf{A}} = \mathbf{T}, \quad \overline{\mathbf{G}} = \mathbf{C}, \quad \overline{\mathbf{C}} = \mathbf{G}, \quad \overline{\mathbf{T}} = \mathbf{A}. \tag{2}$$

More specifically, in a DNA duplex (dDNA), the base **A** on one strand pairs with **T** on the opposite strand by means of two hydrogen bonds, while **C** pairs with **G** by means of three hydrogen bonds (i.e. the strength of the bond between **G** and **C** is stronger than the bond between **A** and **T**). For DNA computing purposes, one is usually concerned with single-stranded DNA (ssDNA) sequences. These sequences are formed by heating DNA double helices to denaturation temperatures, at which they break down into single strands. If the temperature is subsequently reduced, strands with large regions of sequence complementarity can bind back together in a process called *hybridization*. Hybridization is assumed to occur only between complementary base pairs, and this process lies at the core of DNA computing.

As a first approximation, oligonucleotide DNA sequences can be simply viewed as words over a four-letter alphabet $Q = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$, with a prescribed set of given properties. The generic notation for such sequences will be $\mathbf{q} = q_1 q_2 \ldots q_n$, with n indicating the length of the sequences. The WC complement $\overline{\mathbf{q}}$ of a DNA sequence is defined as $\overline{q_1}\,\overline{q_2} \ldots \overline{q_n}$, $\overline{q_i}$ being the WC complement of $\overline{q_i}$ as given by (2). Let $\mathcal{C}$ be a collection of ssDNA strands. The following definitions of distance measures are useful in the context of DNA computing:

$$\begin{aligned} d_H(\mathcal{C}) &= \min_{\substack{\mathbf{p} \neq \mathbf{q},\\ \mathbf{q},\mathbf{p} \in \mathcal{C}}} d_H(\mathbf{p},\mathbf{q}),\\ d^{RC}(\mathcal{C}) &= \min_{\mathbf{q},\mathbf{p} \in \mathcal{C}} d_H(\mathbf{p},\mathbf{q}^{RC}),\\ d_{WC}(\mathbf{p},\mathbf{q}) &= d_H(\mathbf{p},\overline{\mathbf{q}}). \end{aligned} \tag{3}$$

The first distance measure (the Hamming distance) is of interest when evaluating (cross)hybridization properties of DNA words under the assumption of a perfectly rigid DNA backbone. For example, consider two words $3' - \mathbf{AAGCTA} - 5'$ and $3' - \mathbf{ATGCTA} - 5'$ at Hamming distance one from each other. For such two words, the reverse complement of the first word, $3' - \mathbf{TAGCTT} - 5'$, would also have a very large affinity to bind to the second word. In order to prevent such a possibility, one has to impose a minimum Hamming distance constraint ($d_H(\mathcal{C}) \geq d$) on the sequences. On the other hand, in order to prevent unwanted hybridization between two DNA words, one needs to ensure that the reverse-complement distance between words is larger then a prescribed threshold (leading to the introduction of the second distance measure, the reverse-complement Hamming distance). As an example, consider two words $3' - \mathbf{AAGCTA} - 5'$ and $3' - \mathbf{TACCTT} - 5'$. The second word, read-out in reverse order, matches the reverse-complement of the first word in all but one position. Hence, under the given model, the second word is very likely to pair up with the reverse-complement of the first codeword. This introduces the need for imposing a minimum reverse-complement distance constraint ($d^{RC}(\mathcal{C}) \geq d$), as given by (3). The importance of the third measure, termed the Watson-Crick distance, will become apparent after the description of the notion of DNA folding (alternatively, DNA secondary structure formation). The *secondary structure* of a DNA codeword $q_1 q_2 \ldots q_n$ is a set, S, of disjoint pairings between complementary bases (q_i, q_j) with $i < j$. A secondary structure is formed by a chemically active oligonucleotide sequence folding back onto itself due to *self-hybridization*, *i.e.*, hybridization between complementary base pairs belonging to the same sequence. As a consequence of the bending, elaborate spatial structures are formed, the most important components of which are loops (including branching, internal, hairpin and bulge loops), stem helical regions, as well as unstructured single strands.

There exist simple computational techniques, based on dynamic programming methods, that can be used to determine the secondary structure of a DNA sequence. Among these techniques, *Nussinov's folding algorithm* is the most straightforward scheme used [17]. Nussinov's algorithm is based on the assumption that in a DNA sequence $\mathbf{c} = c_1 c_2 \ldots c_n$, the energy between a pair of bases, $\alpha(c_i, c_j)$, is independent of all other pairs. For simplicity, one can assume that $\alpha(c_i, c_j) = -1$ if c_i and c_j are Watson-Crick complements, and $\alpha(c_i, c_j) = 0$ otherwise. Let $E_{i,j}$ denote the minimum free energy of the subsequence $c_i, \ldots, c_j$. The independence assumption allows us to compute the minimum free energy of the sequence $c_1, c_2, \ldots, c_n$ through the recursion

$$E_{i,j} = \min \left\{ \begin{array}{l} E_{i+1,j-1} + \alpha(c_i, c_j), \\ \quad E_{i,k-1} + E_{k,j}, \qquad i < k \leq j, \end{array} \right. \tag{4}$$

where $E_{i,i} = 0$ for $i = 1, 2, ..., n$ and $E_{i,i-1} = 0$ for $i = 2, ..., n$. The value of $E_{1,n}$ is the minimum free energy of a secondary structure of $c_1, c_2, ..., c_n$. Note that $E_{1,n} \leq 0$. A very large negative value for the free energy $E_{1,n}$ of a sequence is a good indicator of the presence of stacked base pairs and loops, *i.e.*, a secondary structure, in the physical DNA sequence. Long loops are known to exhibit destabilizing effects on the secondary structure, leading to unfolding of the strand. The Watson-Crick distance defined above is in some sense a measure of the length of a hairpin loop: if it is equal to its maximum value when evaluated for a sequence and a certain number of its *consecutive* shifts (say N), it indicates that a loop of length N exists in the structure. The larger the value of N, the more unlikely it is for the secondary structure to remain stable - the more likely it is the sequence will not exhibit a secondary structure.

A. Duplex Stability and mRNA Secondary Structures

The stability of a duplex formed by hybridization of short ssDNA strands is usually assessed in terms of the strands free energy. The free energy depends on the particular ssDNA sequences involved - more precisely, it depends on the nearest neighbor interaction energies but not on the composition of the base pairs [3]. For a dDNA with a single strand sequence given by $\mathbf{c}$, the free energy

E_{free} can be accurately approximated as

$$E_{free} = \kappa + \sum_{i=1}^{n-1} \epsilon(c_i, c_{i+1}),$$

where $\epsilon(c_i, c_{i+1})$ denotes the interaction energy between c_i and c_{i+1}, while κ denoting a correction factor which depends on the number of G and C bases in the sequence **c**. A total number of ten different nearest neighbor interactions are possible, and the stabilities of interactions are listed in [3]. The nearest neighbor interaction energies can be used to determine the stability of a specific hybridized duplex, or the minimum length od a duplex required to achieve a certain energy level etc. These problems are, of course, of special interest for controlling gene expressions through post-transcriptional silencing. But there exist very specific constraints imposed in such a setting: hybridization has to be performed with respect to *sub-sequences* of some well-defined mRNA strand. In such a strand, one has to identify the subsequence that gives the most stable hybridization pattern. Furthermore, due to the affinity of mRNA to self-hybridize (i.e. to allow for hybridization between Watson-Crick complementary sub-sequences on the same strand) one also has to know which subsequences of an mRNA strand are un-hybridized. This introduces the need to find simple methods for identifying secondary structures of mRNA and ssDNA strands; these structures also depend on the ordering of the base pairs in the strand.

In the former case, one can use the nearest-neighbor interaction graph to identify: a) *all* sequences of some given length, for which the duplex stability will be sufficiently high (say, greater than a threshold T); b) *the smallest length* for which there exist at least a prescribed number of sequences with stability larger than T. Clearly, such sequences correspond to paths in the nearest-neighbor graph that have vertex cost exceeding T. For example, if the smallest stability level is 20 and the sequences are to be of length five, then **CGCAA**, **GGGGG**, **GGGGT** etc. would satisfy this requirement. But there is no guarantee that the reverse complement of these sequences exist in the targeted biological strand. Furthermore, there is no indication that if such a sub-strand even exists, that it is not "folded" and inaccessible for hybridization. For example, the sequence **TACGCAAAAATTGCGAA** contains the reverse-complement of **CGCAA**, but as part of a stem (helix formation). In the latter case, one can use the Watson-Crick distance metric to try to identify loops in the secondary structure. For example, due to the well-known *no-sharp-turn* constraint, one can focus on the Watson-Crick distance of a targeted sequence and its 1-st,2-nd,..., s-th shift. If all these distances are equal to the lengths of the underlying sequences, then a hairpin loop of length s is very likely to exist within the secondary structure. Bases on the loop can be targeted for hybridization. More details about these and other coding-theoretic problems arising in DNA computing can be found in the companion papers [13] and [14].

Acknowledgment: This work was funded by the NSF Grant CCF-0514921.

References

[1] L.M. Adleman, "Molecular Computation of Solutions to Combinatorial Problems," *Science*, vol. 266, pp. 1021–1024, Nov. 1994.

[2] Y. Benenson1, B. Gil, U. Ben-Dor, R. Adar and E. Shapiro, "An autonomous molecular computer for logical control of gene expression," *Nature*, pp. 1-6, 2004.

[3] K. Breslauer, R. Frank, H. Blocker, and L. Marky, "Predicting DNA Duplex Stability from the Base Sequence," *Proceedings of the National Academy of Science*, USA 83, pp. 3746–3750, 1986.

[4] R. Bundschuh, and T. Hua, Phys. Rev. Lett. 83 (1999) 1479.

[5] D. Chung, T. Richardson, and R Urbanke, "Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes Using Gaussian Approximation," *IEEE Trans. on Inform. Theory*, Vol. 41, No. 2, pp. 657-671, Feb. 2001.

[6] P. Decelles, "Control of Gene Expressions," *Web Resource*,

[7] A. Dyachkov, P. Erdos, A. Macula, V. Rykov, D. Torney, C-S. Tung, P. Vilenkin and S. White "Exordium on DNA Codes," *IEEE Trans. on Inform. Theory*, Vol. 46, No. 4, July 2000.

[8] P. Gaborit, and H. King, "Linear constructions for DNA codes," preprint.

[9] M.A. Gibson, and J. Bruck, "A probabilistic model of a prokaryotic gene and its regulation," *preprint*.

[10] Tommi Jaakkola, Michael I. Jordan, "Computing Upper and Lower Bounds on Likelihoods in Intractable Networks," *Proceedings of the Twelfth Conference of Uncertainty in AI*, Morgan Kauffman, 1996.

[11] H.d Jong, "Project HELIX: Mathematical Modeling of Genetic Regulatory Networks," *preprint*.

[12] O.D. King, "Bounds for DNA codes with constant GC-content," *The Electronic Journal of Combinatorics*, vol. 10, no. 1, #R33, 2003.

[13] O. Milenkovic and N. Kashyap, "New Constructions of Codes for DNA Computing," accepted for presentation at WCC 2005, Bergen, Norway.

[14] O. Milenkovic and N. Kashyap, "Constructions of DNA Codes without Secondary Structure," submitted to ISIT2005, Adelaide, Australia.

[15] S. Mneimneh, "Computational Biology Lecture 20: RNA secondary structures," available online at `engr.smu.edu/~saad/courses/cse8354/lectures/lecture20.pdf`.

[16] A. Montanari and M. Mezard, Phys. Rev. Lett. 86 (2001) 2178.

[17] R. Nussinov, G. Pieczenik, J.R. Griggs and D.J. Kleitman, "Algorithms for loop matchings," *SIAM J. Appl. Math.*, vol. 35, no. 1, pp. 68–82, 1978.

[18] *Time Warps, String Edits, and Macromolecules - the Theory and Practice of Sequence Comparison*, Editors: D. Sankoff and J. Kruskal, CSLI Publications, 1999.

[19] I. Shmulevich, E.R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean Networks: a Rule- Based Uncertainty Model For Gene Regulatory Networks," *Bioinformatics*, 18, pp. 261-274, 2002.

[20] T. Tian and K. Burrage, "Regulatory Neural Network Models for Gene Regulatory Networks," *preprint*.

[21] N. van Kampen, *Stochastic Processes in Physics and Chemistry*, Elsevier, 1997.

[22] R. Wadkins, "Targeting DNA Secondary Structures," *Current Medicinal Chemistry*, Vol. 7, pp. 1-15, 2000.

[23] Z.J. Wang, J. Chen, and K Liu, "Quantitative Modelling of Genetic Regulatory Networks by Incorporating Genomic Data Sources," at http://binary.engin.brown.edu/publication/regNet_GSPS04_v3.pdf.

[24] Ambion - The RNA Company, "RNA Interference and Gene Silencing: History and Overview / A Bizarre Phenomenon is Discovered: Cosuppression and PTGS in Plants," available at http://www.ambion.com/techlib/hottopics/rnai/