

Soft Molecular Computing

Max Garzon
Computer Science
The University of Memphis
Memphis, TN 38152
mgarzon@memphis.edu

John A. Rose
Electrical Engineering
The University of Memphis
Memphis, TN 38152
johnrose@memphis.edu

Russell J. Deaton
Electrical Engineering
The University of Memphis
Memphis, TN 38152
rjdeaton@memphis.edu

Donald R. Franceschetti
Physics
The University of Memphis
Memphis, TN 38152
dfrncsch@memphis.edu

Abstract

Molecular computing (MC) utilizes the complex interaction of biomolecules and molecular biology protocols to effect computation. Lab experiments in MC are unreliable, inefficient, unscalable, and expensive compared to conventional computing standards. A critical issue in MC is therefore to test protocols to minimize errors and mishaps that can thwart experiments when actually run *in vitro*. The purpose of this paper is to describe Edna, an integrated software platform developed to address this problem. The platform will allow MC practitioners to use digital computers to gain insight on the performance of a protocol before it actually unfolds in the tube. Currently, Edna provides tools to find good encodings for a given set of hybridization conditions, by design or evolution, tools to visualize the quality of these encodings, tools to estimate the complexity of given protocols based on bounded complexity, and a virtual test tube simulator based on local interactions between electronic DNA molecules. The virtual tube has allowed us to reproduce Adleman's experiment *in silico*. Edna includes graphical interfaces, click-and-drag facilities, is object-oriented, extensible, and so that it can easily evolve as the field progresses.

1 Introduction

Adleman's ground-breaking work [Adleman, 1994] demonstrated the way to use molecules for computational purposes. Five years later, substantial obstacles remain to bring the potential of molecular computing into the realm of practical and efficient computation, at least in a niche of core applications. Molecular Computing problems are solved in three phases: *encoding* that maps the problem onto DNA strands, *molecular operations* that perform the basic core processing, and *extraction/detection* that makes the results visible to the naked eye. A typical set up consists of a multiset of molecules in a tube that encode the initial conditions of the computation. An algorithmic problem is solved by using a DNA program, herein referred to as a *protocol*, which is defined as a time-ordered sequence of elementary steps given by basic bio-operations (BIOPS) that change the physico-chemical state of the molecules inside the tube.

The decisive impact of errors generated by improper hybridizations during a DNA computation has now been recognized. Because the physical processes involved in molecular computing are much more uncertain, involved, and expensive than digital operations, simulating those processes and protocols beforehand would be advantageous. The situation is analogous to the fabrication of solid-state integrated circuits (IC). IC fabrication facilities are extremely costly to set up. In order to insure success, new chips are commonly simulated from the physical fabrication processes, through the device physics, to logic and circuit level verification. Likewise molecular computing devices could be simulated from the low-level physical processes, such as hybridization reactions, through the molecular biology protocols, to a final verification of the algorithms that they implement. Preliminary work in this direction has centered on the encod-

ing problem (see [Deaton et al., 1998], [Hartemink and Gifford, 1997, Khodor, Hartemink and Gifford, 1997] and [Liu et. al, 1998], for example), because of its decisive influence on the outcome of a computation. Once the encoding of the inputs of a problem have been chosen, the results of a DNA experiment are essentially determined up to laboratory execution. If the encodings are prone to errors, the experiment can be repeated any number of times and always provide an erroneous result. Focus on the massive parallelism and complexity of interactions in the tube, an important source of computational power in molecular computing, however, has prevented other important aspects of the problem from being addressed, such as reliability, efficiency and scalability of the protocols.

The purpose of this paper is to describe **Edna**, an integrated software platform that our group has developed in the last few years to address this problem. The basic assumption is that molecular computing will be effective to address problems beyond the reach of conventional computing. Ideally, one would like to have some sort of virtual tube, much like a wind tunnel for aerodynamics, in which some electronic DNA-like molecules could be tested under variable reaction conditions, and at a much lower price than actual lab experiments. Since such an object is probably nonexistent (more below in the discussion in the conclusion), the next best thing is to create a piece of software, running on solid-state computers, that could provide insight on the actual test tube runs before they are carried out in the lab. In general, the software platform will allow practitioners to determine good encoding molecules for the data in the problem, the best experimental conditions for their protocols, some sense of the reliability of the answers produced, and even perhaps some indication of the complexity of the task as it will unfold in the tube. Currently, **Edna** contains tools to find good encodings for a given set of hybridization conditions, by design or evolution, tools to visualize the quality of these encodings, tools to estimate the reliability of the protocols, and tools to estimate the complexity of given protocols based on bounded complexity [Garzon et al., 1998]. These tools are justified and validated on various analyses of encodings and reactions conditions that are briefly described in the following sections. It is clear that the hypotheses on which the tools have been built will not be agreed upon by all researchers in the field and will necessarily change overtime. Therefore, the platform has been made object-oriented and flexible enough that it can be easily modified and/or extended to incorporate further research on these issues and be upgraded as the field progresses. Although not emphasized below, **Edna** includes graphical interfaces and click-and-drag facilities to enable easy use. Preliminary results indicate that **Edna** may even be potentially useful to solve *in silico* instances of some problems which would be challenging enough for *in vitro* experiments.

2 Encoding Tools

Computational algorithms rely upon physical systems for their implementation. As a result of inherent physical uncertainty, no architecture can ever be entirely error-free. This applies to solid-state electronics, quantum computational devices, and DNA-based computers. A quantitative characterization of the impact of physical uncertainties upon the computational reliability is therefore necessary, whenever a high degree of confidence in the computational result is desired. This problem has been resolved for conventional computation after half a century of solid-state electronics, but remains unsolved for DNA computing. Specifically, the encoding problem for DNA computing consists of mapping the instances of an algorithmic problem in a systematic manner onto specific molecules and chemical protocols so that the resulting reaction products are very likely to contain the answers to the problem's instances. A good solution of the problem prevents unwanted hybridization errors and enables easy retrieval of the answer(s) in the extraction steps.

In the past, two kinds of approaches have been taken to frame and address the encoding problem: heuristic and combinatorial. Usually, heuristic methods to minimize error have been used to encode the problem instances in a set of oligonucleotide codewords [Mir, 1997, Smith, 1996, Deaton et al., 1998, Zhang and Shin, 1998, Liu et. al, 1998]. In each case, the encoding strategy may be useful for minimizing the potential for a particular kind of hybridization error, or for a given specific problem instance and computational architecture. They lack, however, a desirable generalizability to arbitrary computational protocols. A more general, physical measure of error potential has been recently introduced by [Rose et al., 1999], called the computational incoherence, ξ . Being physically based, ξ should generalize encoding to any problem instance or computational architecture. Although a physically-based encoding strategy, such as ξ , may be closer to the ground-truth as far as hybridization error is concerned, its determination can be excessively expensive computationally.

In the combinatorial approach, a measure of hybridization affinity is used to quantify the likelihood that two strands will form a double strand. These measures include Hamming distance [Deaton et al., 1998], [Liu et. al, 1998], and a new hybridization distance introduced in [Garzon et al., 1997]. The latter h-metric appears to capture some of the physical constraints on an encoding in a purely combinatorial measure. Therefore, a combinatorial measure of the tendency to hybridize may have an advantage if computational short-cuts from coding theory or computational biology can be applied, or the structure of the space of all possible encodings (the so-called DNA hypercubes) is known. A tool for finding good encodings in the h-metric space is included in **Edna**.

We present some details on each of these two approaches. Since the Hamming distance ignores frame-shift errors and is not generally appropriate except in special circumstances (such as the Madison group’s codewords), we restrict ourselves to the computational incoherence and the h-metric.

2.1 Computational Incoherence

One physically-based measure that has been used to characterize the hybridization potential of a particular species is the duplex melting temperature, T_m . An examination of the T_m ’s of planned versus unplanned hybrids has been demonstrated to be a useful diagnostic method for the screening [Hartemink and Gifford, 1997] and design [Frutos et al., 1997] of computational sets of molecules. However, T_m is not entirely satisfactory for a fundamental characterization of hybridization error. T_m properly refers to the propensity of stacking for a stochastically observed pair of adjacent base pairs, and not to the overall state of a given duplex. In addition, as the melting curve of any finite length DNA duplex will necessarily have a substantial width, knowledge of T_m is insufficient to predict the state (bound or unbound) of an observed instance of the duplex at arbitrary reaction temperatures. Furthermore, relative rankings in the melting temperatures of two duplexes need not correspond to relative rankings in energetic stability at a given temperature [Cantor & Schimmel, 1980].

An alternative physical measure that addresses some of these drawbacks, the computational incoherence (CI), has been proposed in [Rose et al., 1999]. The annealed ensemble consists thus of numerous instances of many distinct duplexes, in equilibrium with a set of single-strand DNA species. The computational incoherence, ξ , uses the principles of equilibrium statistical mechanics to estimate the ensemble average probability of an error hybridization per hybridization event, given a specified set of planned hybridizations, a set of encodings, and a reaction temperature. Basically, ξ gives the probability of an unplanned hybridization in a collection DNA molecules, and therefore, is a measure of error potential for the ensemble, rather than for pairs of oligonucleotides, as previously has been the case for T_m , Hamming distance, and other error measures. Details of the derivation can be found in [Rose et al., 1999]. What follows is the gist of that argument.

The principles of statistical thermodynamics have been used to characterize the melting behavior of DNA oligonucleotides and polynucleotides [Wartell and Benight, 1985, Paner et al., 1990]. Together with ξ , they are applied to the problem of unplanned hybridizations in a DNA computation. The basic quantity to calculate is the partition function of a duplex that is randomly selected from a tube containing a DNA computation. The partition function, Z , describes the equilibrium distribution of the quantum states of the duplex among the macroscopic configurations of that duplex. The number of quantum states at a given energy of the duplex is given by the Gibbs factor, $Z_j = \exp(-\Delta G_j/RT)$, where ΔG_j is the Gibbs free energy of formation for the duplex, R is the molar gas constant, and T is the absolute temperature. In a DNA computation, the randomly selected duplex will be observed to lie in exactly one of two distinct states: it will be hybridized in a manner either consistent (state \bar{e}), or inconsistent (state e) with the hybridization rules of the computation. The probability that this duplex is in state e is then given by the ratio of the sum of the Gibbs factors for all possible “error” configurations of the duplex and the partition function for the duplex. Therefore, as an ensemble measure of hybridization error, a measure ξ can systematically group all unplanned hybridizations into a single error category. In accordance with the principles of equilibrium statistical mechanics, however, the mixture must be allowed to reach equilibrium prior to the addition of any secondary enzymes (i.e., ligase). This ξ is completely general. The possible configurations of a duplex include the details of both primary and secondary structure. such as bulges, mismatches, and other secondary structures that contribute to a specific configuration [Cantor & Schimmel, 1980]. In addition, concentrations of the reacting species are included through the normal equilibrium expression [Cantor & Schimmel, 1980]. Therefore, the general form for ξ accounts for all possible states of both single- and double-stranded species.

The problem, however, is that in the general form, calculation of ξ is intractable. This is due to several reasons. First, calculation of ξ is only as good as the thermodynamic data available. Measurements of the energetic factors contributing to ΔG_j for base-pair stacking exist [SantaLucia, 1998], as well as for some details of the secondary structure [Marky and Blumenfeld, 1983]. Complete data for all possible structural details under all possible reaction conditions, however, is not available. In addition, all possible configurations of the species present in the reaction must be considered. This number is huge, and analogously to the problem of computing the number of alignments between two sequences in computational biology, it is computationally intractable. Lastly, direct calculation of ξ requires that the concentrations of all species be calculated, which is a difficult problem involving coupled quadratic equations of arbitrary molecularity. Given these factors, approximations must be made in order to reduce ξ to a tractable form.

In [Rose et al., 1999], an approximation ξ' of the computational incoherence ξ has been computed as follows. ΔG_j is defined relative to a zero energy state, which is equal to that of the unstacked, dissociated, single strands [Cantor & Schimmel, 1980], contributions from single stranded regions of the duplex are considered negligible (thus reducing Z to the conformal partition function, Z_c [Cantor & Schimmel, 1980]), and attention is restricted to short polygos (roughly, length ≤ 200 bps). With these assumptions, the model becomes the staggered zipper model [Benight et al., 1981, Cantor & Schimmel, 1980], in which configurations containing internal loops are discarded. The Gibb's factor of each duplex configuration is then related to the total difference in the free energies between the stacked and unstacked states of the duplex. The ΔG_j of duplex j may be estimated by summing the free energy changes of stacking each successive pair of adjacent hydrogen-bonded base pairs $\Delta G_{i,i+1}$ during formation of the duplex from the totally unstacked state, in accordance with the nearest-neighbor model of duplex formation. An estimate of the nearest-neighbor parameters ($\Delta H_{i,i'}^0, \Delta S_{i,i'}^0$) for each of the 10 WC nearest-neighbor pairs has been reported in [SantaLucia, 1998, Allawi and SantaLucia, 1997]. The total free energy of formation of a duplex j is given by

$$\Delta G_j = \sum_i \Delta G_{i,i+1} + \Delta G_{init} + \Delta G_{sym}, \quad (1)$$

where the sum is performed over adjacent pairs of hydrogen bonded base pairs i and $i+1$, ΔG_{init} is the free energy of initiation at each end, and ΔG_{sym} is a symmetry correction. For the annealing reaction in which dissociation is negligible and the initial concentrations of single stranded species are equal, the ensemble average probability that a duplex will be found in an error binding mode is computed as

$$\xi \approx \frac{\sum_{i,j>i} Z_e^{i,j}}{\sum_{i,j>i} Z_c^{i,j}}, \quad (2)$$

where $Z_e^{i,j}$ is the sum of the Gibbs factors of all error configurations and $Z_c^{i,j}$ is the conformal partition function between single-stranded DNA species i and j .

Edna includes a tool to compute this form of ξ of given encodings. Reaction specifications required from the user include the set of graph characteristics (vertex number and edge identities) and the length of encoded single-stranded DNA molecules. Stochastic generation of edges is also supported to allow the assessment of the characteristics of random graphs. Reaction conditions may be set by the user or set automatically to the calculated $T_{reaction}^{max}$. **Edna** can calculate the following information for each assessed encoding:

- The set of T_m and ΔT_m for planned modes of annealings;
- A maximum suggested reaction temperature appropriate for annealing, defined as equal to the minimum member of the set $T_m - \frac{1}{2}\Delta T_m$ over planned modes of annealing;
- ξ' computed for the selected set of reaction conditions;
- An error diagnostic matrix consisting of the set of conditional hybridization error probabilities for each pair of encoding oligos.

In addition, **Edna** implements a standard genetic algorithm for evolving encodings using $-\log \xi'$ as a fitness function. The user can select various parameters of the genetic algorithm, including mutation and crossover rates, population size, crossover point number, and parental number. The initial population of encodings

may be seeded by the user, or may be generated partially or entirely at random. Convergence occurs upon the achievement of either a set maximum number of evolved generations or the emergence of an encoding having a minimally acceptable fitness. The built-in genetic algorithm facility has allowed us to evolve much more coherent encodings for Adleman’s original problem.

2.2 The h-metric

The physico-chemical reality of the tube, particularly framewhifts in hybridization, makes it clear that the Hamming distance is not an adequate measure of hybridization likelihood. Frame shifts appear to be accountable for, nevertheless, by a generalization of the Hamming metric, the so-called *h-metric* [Garzon et al., 1997]. The h-metric seems to capture enough of the reality of reactions conditions and the complexity of test tube hybridizations to frame and adequately solve the encoding problem. Moreover, it is combinatorially-based and computable in quadratic time. The h-measure between two oligos x and y is defined as the minimum of all Hamming distances obtained by successively shifting and lining up the WC-complement of y against x . the h-metric is defined for so-called poligos, namely equivalence classes of n -mers at an h-measure 0 of each other. (This is a technical point required by the fact that the h-measure does not give, strictly speaking, a metric space.) If some shift of one strand somewhat or other matches a segment of the other, the measure is reduced from n by the value of the Hamming distance between the shifted strands. Thus oligos in the same class are WC complementary and will most certainly hybridize. A small measure indicates that the two oligos are likely to stick to each other one way or another; a large measure indicates that *under whatever physico-chemical conditions* y finds itself in the proximity of x , they are far from containing many WC complementary pairs (let alone segments), and are therefore less likely to hybridize, i.e., they are more likely to avoid an erroneous unwanted hybridization. The maximum length of these segments is controlled by a parameter τ , that is a fairly coarse expression of the reaction conditions. Therefore the h-metric can be regarded as measuring the degree of inertia against hybridization.

The resulting metric space of all possible encoding n -mers is called the DNA_n cube of dimension n . A solution of the encoding problem thus consists of finding good codes, in the sense of coding theory, in the higher-dimensional DNA hypercubes, in lieu of the ordinary Hamming boolean hypercubes. They furnish encodings that are, by definition, capable of *preventing* errors them from occurring, largely independently of the type of experiment they are used for. Error-*correcting* approaches have been suggested by several authors (see, for example, [Adleman, 1996, Karp-Kenyon-Waarts, 1996]), but they attempt to detect and correct errors (as is common in information theory), that have been allowed to occur. It appears possible, therefore, to *avoid* as many errors as possible, and perhaps use error-correcting methods to handle errors that cannot be prevented with the encodings generated through the h-metric.

Likewise, **Edna** includes a tool to compute, visualize, manipulate, and evolve encodings for molecular protocols based on the h-metric criterion. The results are integrated so that encodings obtained using one criterion (e.g., h-distance) can be compared against other criteria (e.g. computational incoherence). **Edna** has been instrumental in elucidating the structure of the DNA cubes, which will give information on optimal encodings, and in determining how much of the relevant physical-chemistry is effectively captured by the h-metric.

2.3 Complexity Estimates of Molecular protocols

Preparing for an experiment in molecular computation includes gaining some insight into how the computation will actually go. That means *efficiency* of the reactions (are there enough molecules to produce the desired products? Are they too numerous to stretch reaction times to equilibrium too long? Are the concentrations of each reactant appropriate?) and some indication of the *reliability* and *scalability* of the results obtained from the experiments. Encodings discussed in the previous section provide a minimum guarantee of reliability. Efficiency and scalability have more to do with the actual reactions that are supposed to take place inside the tube, i.e., with the complexity of the molecular computation.

Various concepts have been used in the field to quantify the complexity of a computational protocol in molecular computing. Earlier ones have readily borrowed from Turing complexity and used as resource a count of tube manipulations to be performed. The goodness of a protocol is measured in the traditional way, by the asymptotic behavior of this count as the size of the problems (and hence molecules) approaches

infinity. Since this approach emphasizes events outside the tube and since oligos larger than 20-mers have scarcely been handled easily in the lab for DNA computing, it does not appear to be a very realistic or useful abstraction for analysis of molecular protocols.

An alternative metric, *bounded complexity* (BC), more focused on reactions conditions and the events happening in the interior of the tube, has been introduced in [Garzon et al., 1998]. Due to the fact that there are many different factors involved in each protocol, the complexity is measured in a so-called *TVn* unit, which is a common unit that permits meaningful comparisons between different solutions to the same or even different problems. One *TVn* unit *1h* gives the

$$1h \equiv \text{unit cost per } 1^\circ\text{C change of temperature per one nucleotide per } 1 \mu\text{l volume reaction.}$$

In order to measure the *complexity* of a protocol, one associates a cost to each elementary operation χ . Each operation is implemented as a laboratory procedure, and therefore this cost depends in general on the full state of the tube, i.e. on many factors like temperature, volume of the test tube, number of different DNA molecules present, enzymes used and their cost, and number of nucleotides that are expected to interact within the tube, to name a few. In general, however, we will make the simplifying assumption that the bulk of the cost depends only on a few of these parameters, i.e., we have assumed that operations only depend explicitly on V , T , N and n . The efficiency of a protocol π can be measured by

$$C(\pi) = \sum_{\Psi \in \pi} C_{\oplus}(n(\Psi), N(\Psi), V(\Psi), T(\Psi)),$$

where it is understood that Ψ denotes the successive tubes that arise from the application of preceding operations. **Edna** will include a tool to specify molecular protocols and estimate and compare their efficiency through bounded complexity.

2.4 Virtual Test Tubes

The foregoing sections have discussed events that are expected to occur outside the test tube in preparation for the actual experiments to be carried out in the tube. At this point, it is only natural to ask, why stop here? Why not provide a simulation of the reactions that might actually happen, full with all the random brownian motion and chains of complex molecular interactions that undoubtedly take place in actual test tubes? The sneak preview is worth watching, particularly since it is much less costly than the actual test tube run.

Edna includes tools to perform a simulation and analysis of test tube experiments based on the measures presented in the previous sections. The simulation is based on purely local molecular interactions between cells in a cellular automaton-like space. The cells represent quanta of 2D or 3D space that may be empty or occupied by nucleotides, solution molecules, and/or other reactants. Each cell is also characterized by associated parameters that render the tube conditions in a realistic way, such as temperature, salinity, covalent bonds, etc. The cellular space can be set up to boundary conditions that reflect tube walls. The transitions of the automaton implement the massive random motion of single stranded molecules compounded over space and time, their multiple attempts and eventual success at hybridizing into double stranded molecules according to various hybridization rules, under various temperatures and reaction conditions and for a variety of input encodings. The virtual tube comes to equilibrium under the Gibbs free energy function, showing a possible outcome of the actual experiment. That this asymptotic behavior has something to do with the actual test tube run can be justified on several grounds. After all, the underlying physical chemistry is based on fairly well known *local* interactions among the molecules and between them and their tube environment. What is unknown, and presumably very powerful, is the compounded effect of all the massive number of interactions in the real tube. A virtual test tube run represents a Montecarlo sample of that effect. Second, experiments that have been actually performed in the lab, such as Adleman's experiment, have been run on **Edna**. Their outcomes may be compared with the known outcomes to benchmark the reliability of the virtual tube and the analyses on which is has been built. The degree of reliability of the virtual test tube can thus be appropriately validated. Preliminary results on the HPP show that **Edna** can be a powerful tool to debug, and in many cases even perform, molecular protocols with high reliability. A systematic study is under way and will be reported elsewhere.

2.5 Discussion and Conclusions

The approach in this paper has been to explore the contribution that silicon-based computers can make to the success of biomolecule-based computing. This approach is justified for several reasons. First, as a new computing paradigm, molecular computing must be evaluated against the performance of conventional solid-state based computing. The impressive progress rates and proven reliability of the latter has led many practitioners to believe that molecular computing can only compete advantageously in a niche of 'killer' applications. Second, basic obstacles remain in the way of bringing the potential of molecular computing to the realm of practical, efficient and reliable computations. A critical one is that molecular computing (MC) utilizes the complex interaction of biomolecules and protocols based on biotechnological developments to perform computation. Although biotechnology to manipulate molecules has been developed to satisfactory standards for biological purposes, they are far from meeting analogous standards in conventional computing media. In particular, molecular computing lab runs in the test tube are too complicated, unreliable, inefficient, and expensive when compared to ordinary runs of conventional software. A re-evaluation of the models on which such a technology is based is required.

Edna, a software platform, has been developed to address the basic problems of encoding, reliability, efficiency and scalability for molecular protocols using DNA molecules. The platform will allow MC practitioners to take advantage of digital computers to gain realistic insights on actual test tube performance of a protocol before it actually unfolds in the tube. **Edna** is object-oriented and extensible, so that it can easily evolve as the field progresses. **Edna** is therefore a research tool that makes it possible to use the advantages of conventional computing to bring to molecular computing comparable levels of reliability and efficiency.

Finally, **Edna** poses a number of interesting questions directly addressing what is perhaps the critical issue in MC: is it the case that there exist elemental unknown computations happening deep inside biomolecule ensembles, powerful enough to propel molecular computing beyond the confines of conventional computers? Or, is it the case that the results of molecular computing could be achieved, in principle, by some sort of electronic DNA, but at such an enormous computational expense to be impossible in real life due to the sheer massive number/volume of molecules involved and/or the inability of current electronics to make more efficient use of the much smaller electrons? Preliminary estimates based on extrapolations of a distributed version of **Edna** running over computer networks (which in fact would create a virtual test tube the size of the internet) give some evidence that even in the latter case, Adleman's experiment would be somewhat of a challenge.

References

- [Allawi and SantaLucia, 1997] H. T. Allawi, J. SantaLucia, Jr., Thermodynamics and NMR of Internal G*T Mismatches in DNA. *Biochemistry* **36**, 10581 (1997).
- [Adleman, 1994] L. M. Adleman, *Science*, **266**, 1021 (1994).
- [Adleman, 1996] L.M. Adleman (1996) On Constructing a Molecular Computer. In [Lipton-Baum, 1996], 1-21.
- [Benight et al., 1981] A. Benight, R. Wartell, and D. Howell, Theory agrees with experimental thermal denaturation of short DNA restriction fragments. *Nature* **289**, 15 (1981).
- [Cantor & Schimmel, 1980] C. R. Cantor, P. R. Schimmel, *Biophysical Chemistry, Part III: The Behavior of Biological Macromolecules* (Freeman, New York, 1980).
- [Deaton et al., 1998] R. Deaton, M. Garzon, R. E. Murphy, J. A. Rose, D. R. Franceschetti, S. E. Stevens, Jr., *Physical Review Letters*, **80**, 417 (1998).
- [Frutos et al., 1997] A. Frutos, Q. Liu, A. Thiel, A. Sanner, A. Condon, L. Smith, R. Corn, Demonstration of a Word Design Strategy for DNA Computing on Surfaces. *Nucleic Acids Research* **25**, 4748 (1997).
- [Garzon et al., 1997] M. Garzon, P. Neathery, R. Deaton, R.C. Murphy, D.R. Franceschetti, S.E. Stevens, Jr.. A New Metric for DNA Computing. In [Koza et al, 1997], 472-478.

- [Garzon et al., 1998] M. Garzon, N. Jonoska, S. Karl. The Bounded Complexity of DNA Computing. Proc. of the 4th workshop, Princeton University, 1998. In [Rubin and Wood, 1999] and special issue of *Biosystems*.
- [Hartemink and Gifford, 1997] A. J. Hartemink, D. K. Gifford, Thermodynamic Simulation of Deoxyoligonucleotide Hybridization of DNA Computation, in [Landweber and Baum, 1999].
- [Khodor, Hartemink and Gifford, 1997] J. Khodor, D.K. Gifford, A. Hartemink (1998), Design and Implementation of Computational Systems Based on Programmed mutagenesis. In [Rubin and Wood, 1999], 101-107; 287-297.
- [Karp-Kenyon-Waarts, 1996] R. Karp, C. Kenyon, O. Waarts (1996), Error-resilient DNA Computation. Proc. 7th Annual Symposium on Discrete Algorithms, SODA 458-467.
- [Koza et al, 1997] J.R. Koza, K. Deb, M. Dorigo, D.B. Fogel, M. Garzon, H. Iba, R.L. Riolo, eds. (1997). *Proc. 2nd Annual Genetic Programming Conference*, Morgan Kaufmann.
- [Koza et al., 1998] J.R. Koza, K. Deb, M. Dorigo, D.B. Fogel, M. Garzon, H. Iba, R.L. Riolo, eds. (1998). *Proc. 3rd Annual Genetic Programming Conference*, Morgan Kaufmann.
- [Landweber and Baum, 1999] L.F. Landweber, E.B. Baum (eds.), *DNA Based Computers II*, Proc. of the 2nd workshop, Princeton University, 1996. DIMACS series of the American Mathematical Society, vol. 44 (1999), 247-258, Providence RI.
- [Lipton-Baum, 1996] R. Lipton, E. Baum (eds.) *DNA Based Computers*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. **27**. American Mathematical Society, Providence RI.
- [Marky and Blumenfeld, 1983] L. Marky, K. Blumenfeld, S. Kozlowski, and K. J. Breslauer, Salt-Dependent Conformational Transitions in the Self-Complementary Deoxydodecanucleotide d(CGCAATTCGCG): Evidence for Hairpin Formation. *Biopolymers*, **22**, 1247-1257 (1983).
- [Mir, 1997] K. U. Mir, *DNA Based Computers II, Princeton University, 1999*, DIMACS Proc. Series (American Mathematical Society, Providence, RI, 1997).
- [Paner et al., 1990] T. Paner, M. Amaratunga, M. Doktycz, A. Benight, Analysis of Melting Transitions of the DNA Hairpins formed from the Oligomer Sequences d[GGATAC(X)₄GTATCC](X = A, T, G, C). *Biopolymers* **29**, 1715(1990).
- [Record and Lohman, 1978] M. T. Record, T. Lohman, A Semiempirical Extension of Polyelectrolyte Theory to the Treatment of Oligoelectrolytes: Application to Oligonucleotide Helix-Coil Transitions. *Biopolymers* **17**, 159 (1978).
- [Rose et al., 1999] J. A. Rose, R. Deaton, D. R. Franceschetti, M. Garzon, S. E. Stevens, Jr., A Statistical Mechanical Treatment of Error in the Annealing Biostep of DNA Computation. Special program in DNA and Molecular Computing at the Genetic and Evolutionary Computation Conference (GECCO-99), Orlando FL, July 13-17, 1999. Morgan-Kaufmann, 1829-1834.
- [Ross et al., 1991] P. Ross, F. Howard, M. Lewis, Thermodynamics of Antiparallel Hairpin-Double Helix Equilibria in DNA Oligonucleotides from Equilibrium Ultracentrifugation. *Biochemistry* **30**, 6269 (1991).
- [Rubin and Wood, 1999] H. Rubin. D. Wood (eds.), 4th DIMACS workshop on DNA Computers. Proc. of the 4th workshop, University of Pennsylvania, 1998. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. **48**. American Mathematical Society, Providence RI.
- [SantaLucia, 1998] J. SantaLucia, Jr., A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics *Proc. Natl. Acad. Sci.* **95**, 1460 (1998).
- [Smith, 1996] W. D. Smith, *DNA Based Computers, Princeton University, 1996*, DIMACS Proc. Series (American Mathematical Society, Providence, RI, 1996).

- [Liu et. al, 1998] Progress toward Demonstration of a Surface-based DNA Computation. In [Koza et al., 1998], 709-717.
- [Vesnaver and Breslauer, 1991] G. Vesnaver, K. Breslauer, The contribution of DNA single-stranded order to the thermodynamics of duplex formation. *Proc. Natl. Acad. Sci.* **88**, 3569 (1991).
- [Wartell and Benight, 1985] R. M. Wartell, A. S. Benight, Thermal Denaturation of DNA Molecules: A Comparison of Theory with Experiment. *PHYSICS REPORTS (Review Section of Physics Letters)* **126**, 67 (1985).
- [Wetmur, 1997] J. G. Wetmur, *Preliminary Proceedings of the Third Annual Meeting on DNA Based Computers, University of Pennsylvania, 1997*, DIMACS Proc. Series (American Mathematical Society, Providence, RI, 1997).
- [Zhang and Shin, 1998] B-T Zhang, S-Y Shin, Molecular Algorithms for Efficient and Reliable DNA Computing. In [Koza et al., 1998], 735-742.