ELSEVIER

BioSystems 45 (1998) 59–66

BioSystems

DNA as a vehicle for the self-assembly model of computing

Michael Conrad *, Klaus-Peter Zauner

Department of Computer Science, Wayne State University, Detroit, MI 48202, USA

Received 30 April 1997; received in revised form 8 July 1997; accepted 22 August 1997

Abstract

A DNA version of the self-assembly model of computing, feasible using currently available laboratory techniques, is proposed. Input signals are coded into unmethylated and methylated oligonucleotides which then hybridize with a backbone that contains complementary sequences. Different input signal patterns are thus represented as DNA duplexes with distinctly different conformational dynamics, in particular different equilibria of *B* and *Z* DNA. The pattern classification activity of the system is mediated by the interactions that lead to the secondary structural organization. Circular dichroism may be used for readout. © 1998 Elsevier Science Ireland Ltd.

Keywords: DNA computing; Self-assembly; Conformational dynamics; Pattern recognition; Molecular computing

1. Introduction

The recent period has seen high interest in the possibility of using DNA as a basis for formal models of computing. The first proposal along this line was made by Vaintsvaig and Liberman (1973), who demonstrated (prior to the discovery of RNA processing) that enzymatic alterations of DNA could yield universal computation. Subsequently this was referred to as DNA word processing (Conrad and Liberman, 1982), with the term ‘word’ motivated by formal language theory. (Word processors in the modern sense were not yet in common use). The recent interest was triggered by Adleman’s recognition and experimental demonstration that the PCR technique allows for a formal model of DNA computing that affords massively powerful fine grained parallelism (Adleman, 1994).

So far the Adleman system has only been applied to small problems, due to technical limitations. Nevertheless, it has opened up a new picture of formal computing based on complex pattern matching operations acting on strings. The system can be viewed from a structural point of view (Conrad and Zauner, 1995; Zauner and Conrad, 1996). A large number of DNA structures are created; the formation of certain specific structures yields a particular hybrid that corre-

* Corresponding author.
E-mail: biocomputing@cs.wayne.edu

PII S0303-2647(97)00062-2

sponds to a solution. Effects depending on DNA secondary structure (or conformational features) must be carefully avoided to preserve a precise correspondence between the wet and paper chemistry. It is this correspondence that makes Adleman type schemes programmable. The salient point is that it is possible to use a simple set of well defined rules to prescribe, with adequate precision, the relevant behavior of the components.

Programmability, however, entails an in-principle cost in terms of computational efficiency (Conrad, 1985, 1988). That the paper chemistry is sufficient to capture all essential features of the wet chemistry means that a vast number of interactions potentially useful for problem solving must either be frozen out or admitted without being exploited. Clearly all interactions that could lead to problematic conformations would have to be excluded. Interactions leading to conformational effects that do interfere with programmability constitute a vast unutilized computational resource.

The purpose of this note is to show that it is possible to capture these side interactions if DNA is contemplated in terms of a computational model more natural to biological materials. The self-assembly model of computing (Conrad, 1990, 1992) captures the key idea, namely that macromolecules such as proteins and DNA use their conformational dynamics to fuse input milieu patterns impinging on them. The behavior of the components is too dependent on the state of the whole system to allow for formal programmability. Evolutionary methods of adaptation are possible, however. As in nature the conformational dynamics can be molded for specific function (‘programmed’) through variation and selection.

2. Self-assembly model

First let us briefly review the self-assembly concept (illustrated schematically in Fig. 1). External signals (say light signals or electrical pulses) arriving along different input lines release differently shaped macromolecules. The input signal pattern will thus be represented by a particular pattern of conformations. These then self-assemble, in the fashion of jigsaw puzzle pieces, to yield a polymacromolecular complex. Shape features common to different complexes will represent different groupings of the possible input patterns. Readout enzymes that recognize these shape features could then be used to produce an output. In this way a symbolic pattern recognition problem is converted to a self-assembly process, thus essentially converted to a process of free energy minimization. All the clever physics of macromolecular self-assembly are brought to bear on the pattern recognition problem. The device may be thought of as crystallizing a solution of the problem.

The molecular shapes released need not be separate molecules. They could be conformational changes triggered in a macromolecule or supermacromolecular complex. Self-assembly would then reduce to the conformational reorganization of the molecule or complex following exposure to the pattern of milieu signals. The recognition capabilities inherent in conformational dynamics has its origin in the highly nonlinear interactions among numerous electrons and atomic nuclei. Complex recurrent networks of conventional switching elements would in general be required to recognize the same class of patterns.

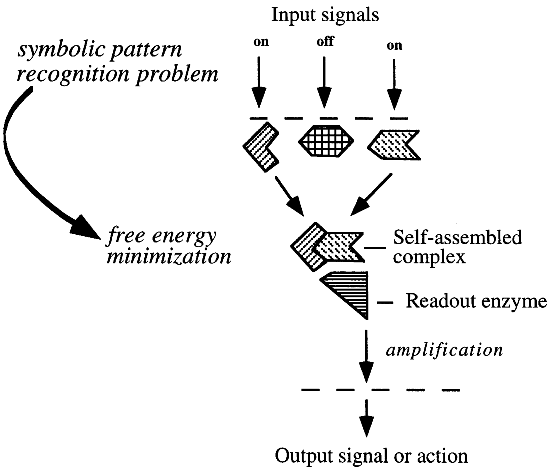

Fig. 1. Self-assembly model of computing. Input signals arriving along different lines are coded into molecular shapes, which then self-assemble to form a complex whose shape features correlate with different groupings of the input patterns. Enzymes recognizing these shape features trigger the output signal.

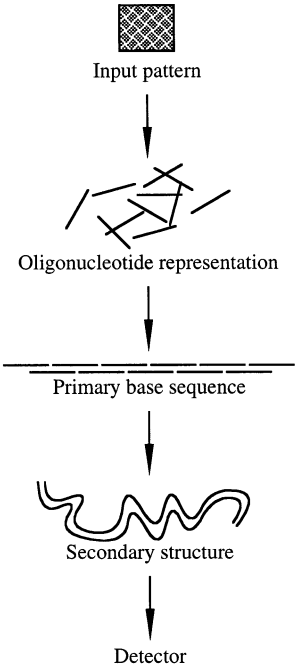

Fig. 2. General scheme of a DNA conformational processor. Input signal patterns are re-represented as collections of different oligonucleotide sequences. These hybridize to form duplexes in a predictable manner, with the number of different types of duplexes depending on the initial choice of oligonucleotides. The formation of the DNA secondary structures from these duplexes is the essential computation step. The resulting conformational features are used to classify the input patterns.

3. Self-assembly computing with DNA

Fig. 2 illustrates the self-assembly model of computing with DNA. The scheme uses linear string matching (or hybridization) to associate different pieces of input information and uses secondary structural effects (or conformational processing) to fuse this information. Different input signals are coded into specific oligonucleotide sequences. These self-assemble through hybridization to yield a primary base sequence. Up to this point the process is largely programmable (i.e. the paper chemistry is sufficient to predict the hybridization products). The primary base sequences determine the range of possible secondary structures (or conformations) of these hybridization products. Given environmental conditions will favor particular conformations in this range. At this point the process is no longer programmable in the formal sense. Many interactions contribute to the development of these conformations, including interactions that link local features of the DNA duplex to its larger scale organization. This development is the main computational process in our model, since the resulting conformational features ultimately depend on the initial signal pattern. The output signal elicited from the secondary structure can thus be used to group different input signal patterns.

The above general scheme allows for a number of feasible realizations. Here we consider an example which utilizes the fact that DNA can switch between right and left handed secondary structures (Sasisekharan and Brahmachari, 1981; Sasisekharan, 1983). The right (*B*) and left (*Z*) forms are in an equilibrium, but the former is strongly favored under common conditions. Base sequence determines whether a stable *Z* form is possible at all; whether this form is actually assumed depends on environmental factors (Rich et al., 1984). Sequences containing alternate purine-pyrimidine residues are more likely to form *Z* DNA. Most importantly for the present purposes, modification of the cytosine through methylation can also stabilize the *Z* DNA form (Zacharias, 1993). Pertinent environmental factors include pH, salt concentration, temperature, solvent properties, and interactions with proteins. *B* and *Z* forms can occur concurrently in the same DNA molecules, in which case they are separated by junctions (*BZ* regions). The utility of *B* and *Z* secondary conformational effects is due to the ease of measuring them through various spectroscopic techniques.

The complete setup is illustrated in Fig. 3. Input signals arrive along different input lines, each line being in one of two states (denoted by 0 or 1). A different oligonucleotide sequence is associated with each input line. If the input signal along a particular line is 0 an unmethylated version of the base sequence associated with this input line is released. If the input signal is 1 then a methylated version is released. Thus 1's are represented by methylated sequences and 0's by corresponding unmethylated sequences. These input channel sequences then hybridize with a back-

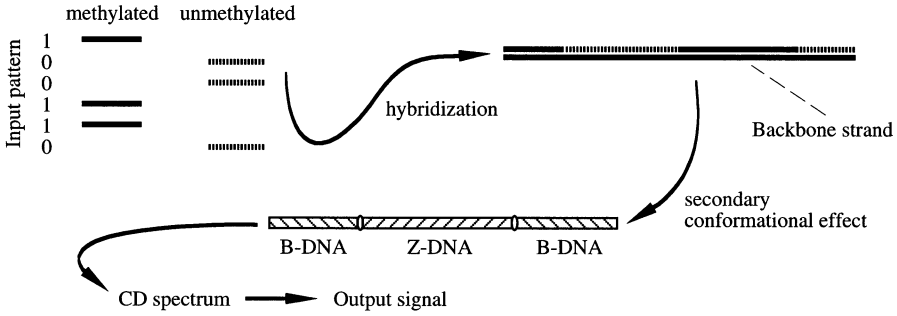

Fig. 3. DNA version of the self-assembly model. Input signals arrive along different input channels, each associated with a specific oligonucleotide sequence. If the channel signal is 1 the corresponding oligonucleotide is methylated; if it is 0 it is not methylated. The oligonucleotides are then allowed to hybridize with a single stranded (backbone) DNA molecule that contains sequences complementary to those associated with the input channels. The resulting hybrid (which may be fixed through ligation) will exhibit a ratio of *Z* to *B* DNA that depends on the input pattern. This ratio, determined through circular dichroism (CD), is used to produce the output signal.

bone molecule (to use the terminology of Roweis et al., 1996) that contains sequences complementary to them. The lined up channel sequences might then be covalently linked by a ligase reaction. Different patterns of methylation will give rise to different secondary conformational effects, in particular formation of *Z* DNA. The whole duplex is then subjected to a definite environment that will determine which of the potential conformational patterns will develop. The extent to which *Z* DNA forms is conveniently detected through circular dichroism (Kennard and Salisbury, 1995). This effect is due to the two components of circularly polarized light being transmitted differentially by the left handed *Z* and right handed *B* forms of DNA. The circular dichroism (CD) spectrum is then used to produce an output signal that classifies the input pattern.

Adaptive techniques are required to train the system to perform a particular desired classification. Evolutionary methods can be applied at any of four stages. The sequences associated with the input channels can be varied. The useful sequences would still have to have complements on the backbone. This reinterpretation of the backbone sequence would yield variant conformational effects, and therefore different groupings of the input patterns. The second point of variation is the backbone sequence itself. This includes changing the possibilities for the input channel sequences or changing the order of their complements on the backbone. It is also possible to change the number of complementary sites on the backbone that are available for a given input sequence. Such reorderings and repetitions would in effect control the interactions among the input sequences. The interaction between two input signals would likely be stronger if their complementary sequences are closer together. The contribution of an input would be amplified if its complementary sequence is repeated. The milieu is the third target of variation. Environmental factors can influence the equilibrium between *B* and *Z* forms in a nonlinear fashion, and consequently altering them would affect the pattern grouping functionality. Feedback learning could also be introduced at this stage by directly coupling a performance measure on the output signal to the composition of the milieu.

The fourth point of adaptation is in the reading of the CD spectrum. The strength of the signal used to decide whether the duplex is to be classified as sufficiently *Z* or sufficiently *B* is itself a variable parameter. In the original self-assembly model (Fig. 1) the interpretation of the self-assembled complex is done at the molecular level by readout enzymes, which then initiate reactions leading to a macroscopic output signal. Such molecular level readouts would allow responses to much more specific features than a simple equilibrium of *B* and *Z* forms.

So far we have not considered the inherent parallelism of the DNA self-assembly processor. This is of critical importance since it yields an enormous scale-up of the input bandwidth. The key point is that the coding of input lines is distinct from the coding of the signals on these lines. The former are coded by base sequence, whereas the latter are coded by the methylation state of at least some of the cytosines in this sequence. The sequence of bases tag the input signal lines that release them in the same fashion as the frequency of a carrier wave tags a radio signal. Methylation, following this analogy, corresponds to modulation of the carrier.

The set of base sequences available for input lines grows as 4^L-3^L, where L is the length of the sequences and 3^L is the number of sequences not containing cytosine. Lengths between 10 and 100 are quite reasonable. A small fraction of these might not support sufficient methylation. Also it would be undesirable to overly commit the sequence space to input lines, since this would close off possibilities for evolutionary variation. But even with these restrictions the number of possible input lines (the fan-in) is enormous as compared to that which is possible with current technical systems.

The set of channel sequences and the backbone interact in parallel. In effect, this is a parallel computational search for the self-assembled structure whose conformation will determine the output signal. This would be the case even if each channel were represented by a single molecule. But realistically each input channel sequence would be represented by a large number of identical molecules. This number could easily be in the hundreds of thousands without creating an undue burden in terms of the total amount of DNA. Micromolar concentrations are reasonable. For example, if the DNA self-assembly processor were asked to manage signal patterns arriving along 10^4 input lines this would allow each of these lines to be represented by 10^9 DNA molecules (taking into account the backbone). Ideally this would lead to the self-assembly of 10^9 duplexes each of whose conformations represents the solution of the pattern recognition problem. But not all of these duplexes need form in order to obtain a correct result. This is due to the fact that the yield of the hybridization phase is part of the adaptation process. A smaller yield would just mean weaker signals in the CD spectrum. The numbers assumed in this example would be equivalent to 500 pmol DNA for all signal sequences in 100 μl of reaction solution. These are quantities typical of standard protocols (Harwood, 1996).

Self-assembly computing utilizes parallelism in a very different way than Adleman type schemes. The new feature is that the search space is explored by the different pathways of conformational self-organization subsequent to duplex formation, whereas in the Adleman scheme they are explored by the number of hybridization pathways. The latter in general generates an exponentially increasing number of reaction products. The number of sequences is initially small but increases as the reaction proceeds. This is because the search is implemented by a branching reaction. In this way fine grained parallelism is used to exhaustively search the space of potential solutions. By contrast, the number of DNA molecules decreases as the computation performed by the self-assembly processor proceeds. This is because all the input sequences aggregate with the backbone to yield a single species of double stranded DNA. The important distinction between the two models is this: the Adleman scheme draws its computational power from the sequence matching aspect of hybridization, whereas the self-assembly scheme draws its power from the conformational dynamics concomitant to this sequence matching.

We can finally consider how the amount of DNA required by the self-assembly processor scales with problem size. For a given pattern classification problem the search space increases as 2^n, where n is the number of signal lines. This space is explored by 2^n pathways for reaching the final conformation (or, more accurately, families of pathways). The number of final detailed conformations should also be 2^n. But in general the number of conformations defined in terms of B and Z regions would be much smaller. The number of possible outputs would be reduced to two in the final reading by the CD spectrometer. Since the computation is mediated by the conformational dynamics the amount of DNA required would not grow exponentially with the size of the pattern recognition problem. However, we should note that the problem of adapting a pattern classifier for arbitrary grouping of input patterns grows as 2 to the 2^n. This is intractable for any known model of computation. Systems with different types of dynamics can address different subclasses of this problem.

4. To RNA and protein

The DNA conformational processor may be viewed as a step towards self-assembly computing. The advantage is that it should be relatively easy to implement given currently available methods. However, the model presented (Fig. 2) does not support all the features of the general self-assembly model (Fig. 1) as efficiently as would RNA and proteins. DNA's primary natural function of information storage limits the richness of its conformational dynamics. The more active functions of RNA and protein, and the fact that linearly distant monomers can be brought into close proximity through the folding process, allows for a much larger variety of specific shapes.

RNA affords some of the advantages of both DNA and proteins. Like DNA it supports predictable hybridization. But in contrast to DNA, the repertoire of possible folded shapes and concomitant functionalities is large. Catalytic RNA's (ribozymes) have been adapted for new functions through in vitro evolution (Joyce, 1989; Robertson and Joyce, 1990). These are advantages from the point of view of conformation-driven computing. For example, it would be possible to realize the general scheme illustrated in Fig. 2 by using RNA oligonucleotides to encode inputs. These would form double stranded hybridization products specific to the input pattern. A subsequent ligation would fix these products as covalent bonded molecules. The new feature would be a melting of the double strand to yield two complementary single strands that when separated will fold to form a 3D shape. The output could be obtained by detecting the folded shape (e.g. with antibodies) or by using it to catalyze a reaction. In this paper we have focused on DNA, however, since working with RNA requires special precautions to prevent contamination with ubiquitous RNases.

Proteins afford yet richer conformational possibilities. The hybridization step of the DNA and RNA realizations is no longer available in this case. The coding of inputs into conformations can proceed in either of two ways. As in Fig. 1 each input signal line can control the release of a protein with a specific shape. Alternately, the input signals could be coded into milieu features which control different conformational properties of a single protein, a complex, or of a collection of proteins that would then assemble to form a larger complex. The confomational dynamics serves to fuse the input signals.

The biological cell utilizes DNA, RNA, and protein in ways that are most appropriate to their different physio-chemical qualities. The information storage function of DNA requires that many possible sequences be functionally equivalent. But such equivalence places restrictions on the variety of conformational dynamics. Proteins carry out specific catalytic and structural functions. Even slight changes in amino acid sequence can alter these functions. The RNA class of molecules provides a compromise between DNA and protein capabilities, and accordingly serves, in modern cells at least, to bridge information storage and active functions. Hybrid conformational processors (such as DNA–RNA hybrids) could facilitate transitions between different classes of materials. Hybrid schemes could also afford computational synergies, as they do in biological cells, but the technical complexity of implementing them for this purpose would increase substantially.

Does the DNA conformational processor described here capture processes that are intrinsic to biology or is it a contrivance that utilizes properties of biological molecules in an unbiological way? Cellular DNA can in a very definite sense be viewed as a pattern recognizer. Different patterning of DNA expression in response to different patterns of milieu signals is the essence of cell differentiation. Methylation probably contributes to this pattern recognition-expression capability. But clearly our use of natural DNA properties in the design proposed here distorts the manner in which these properties are used in biological cells. The macroscopic expression of conformational effects in cells and organisms percolates to the macro level through a chain of amplification processes. The CD mechanism of readout is obviously much simpler, since it occurs in a single step.

5. Further remarks

The DNA model would utilize fewer interactions than RNA and especially protein realizations, but have the advantage of being closer to realization in the laboratory. We regard the DNA model as providing a pathway to potentially more powerful realizations with RNA and eventually protein. Following this pathway would mean trading in more and more of the programmability for computational power. The DNA and RNA models are programmable so far as hybridization is concerned, but nonprogrammable with respect to conformational dynamics. A learning or evolutionary paradigm is appropriate, since the conformational dynamics must be adapted to the imposed pattern processing task. This direction is orthogonal to DNA computing models that implement formal string processing operations. The latter, in their very nature, trade away the vast majority of potential interactions for prescriptive control. They also inherit the rigidity of formal computing systems and hence are less well suited to adaptive approaches.

The DNA conformational processor described here, and also RNA and protein variants, are directed to what on the surface may appear to be a rather specific type of computing, namely pattern recognition. But in fact, the components from which all present day general purpose computers are built up are pattern recognizers. The difference is that the components used (e.g. NAND gates) recognize extremely simple, rigid patterns, whereas conformational processors are most suitable for complex, ambiguous patterns. In principle, it would be possible to construct a network of conformational processors that would be programmable at the interpretative level and that would therefore have general powers of computation (Conrad, 1985). The human brain is presumably such an example. But the natural application domains of conformational processors, like those of the brain, would most plausibly take advantage of the efficiency and adaptability advantages of nonprogrammable computing to address problems that are complementary to those at which digital machines and other formal models of computation excel. We can envisage, in the future, that conventional machines will be augmented by molecular co-processors operating on the conformational principle.

Acknowledgements

This paper is based on research supported by NSF Grant No. ECS-9409780. We thank Svetlana Morosyuk for helpful discussion.

References

Adleman, L.M., 1994. Molecular computation of solutions to combinatorial problems. Science 266, 1021–1024.

Conrad, M., 1985. On design principles for a molecular computer. Commun. ACM 28, 464–480.

Conrad, M., 1988. The price of programmability. In: Herken, R. (Ed.), The Universal Turing Machine: a Half-Century Survey. Kammerer and Unverzagt, Hamburg, pp. 285–307.

Conrad, M., 1990. Molecular computing. In: Yovits, M. (Ed.), Advances in Computers. Academic Press, New York, pp. 235–324.

Conrad, M., 1992. Molecular computing: the lock-key paradigm. Computer 25 (11), 11–20.

Conrad, M., Liberman, E.A., 1982. Molecular computing as a link between biological and physical theory. J. Theor. Biol. 98, 239–252.

Conrad, M., Zauner, K.-P., 1995. Molecular computing: steps toward integration. Oyo Buturi (Jpn. Soc. Appl. Phys.) 64, 1002–1006.

Harwood, A.J., 1996. Basic DNA and RNA protocols. In: Methods in Molecular Biology, vol. 58. Humana Press, Totowa, .

Joyce, G.F., 1989. RNA evolution and the origins of life. Nature 338, 217–224.

Kennard, O., Salisbury, S.A., 1995. DNA structure. In: Meyer, R.A. (Ed.), Molecular Biology and Biotechnology. VCH, New York, pp. 242–247.

Rich, A., Nordheim, A., Wang, A.H.-J., 1984. The chemistry and biology of left-handed *Z*-DNA. Ann. Rev. Biochem. 53, 791–846.

Robertson, D.L., Joyce, G.F., 1990. Selection in vitro of an RNA enzyme that specifically cleaves single-stranded DNA. Nature 344, 467–468.

Roweis, S., Winfree, E., Burgoyne, R., Chelyapov, N.V., Goodman, M.F., Rothemund, P.W.K, Adleman, L.M., 1996. A sticker based model for DNA computation. Presented at the 2nd Ann. Meet. on DNA Based Computing. Princeton University, 10–12 June.

Sasisekharan, V., Brahmachari, S.K., 1981. *B* to *Z* transitions in DNA fibre: the question of handedness of the duplex. Curr. Sci. 50, 10–13.

Sasisekharan, V., 1983. Left-handed DNA duplexes. Cold Spring Harbor Symp. Quant. Biol. 47, 45–52.

Vaintsvaig, M.N., Liberman, E.A., 1973. Formal description of cell molecular computer. Biofizika 18, 939–942.

Zacharias W., 1993. Methylation of cytosine influences the DNA structure. In: Jost, J.P., Saluz, H.P. (Eds.), DNA Methylation: Molecular Biology and Biological Significance. Birkhäuser Verlag, Basel, pp. 27–38.

Zauner, K.-P., Conrad, M., 1996. Parallel computing with DNA: toward the anti-universal machine. In: Voigt, H.M. Ebeling, W., Rechenberg, I., Schwefel, H.P. (Eds.), Parallel Problem Solving from Nature—PPSN IV, Lecture Notes in Computer Science, vol. 1141. Springer-Verlag, Berlin, pp. 696–705.