DNA Extraction by XPCR

Giuditta Franco[1], Cinzia Giagulli[2], Carlo Laudanna[2], and Vincenzo Manca[1]

[1] Department of Computer Science, University of Verona
franco@sci.univr.it, vincenzo.manca@univr.it
[2] Section of General Pathology,
Department of Pathology, University of Verona
{cinzia.giagulli, carlo.laudanna}@univr.it

Abstract. The extraction of DNA strands including a given sequence of bases is a crucial step in the Adleman-Lipton extract model of DNA computing. In this paper, a special type of PCR is presented with a related algorithm which performs a specified extraction from a given pool of DNA double stranded (shortly dsDNA) molecules. This kind of PCR, called Cross Pairing PCR (shortly XPCR) was tested in several situations, and used in a biotechnological procedure which implements the extraction algorithm.

1 Introduction

Since the introduction of the Adleman-Lipton extract model [1, 8], the fundamental schema of a DNA algorithm, for solving an instance of a combinatorial problem, is the following: i) Generation of a pool DNA strands encoding all possible solutions (the solution space), ii) Extraction of those that are the true solutions of the given instance. This second step is performed by a sequence of elementary extraction sub-steps, where at each sub-step all the strands where a specific sub-strand occurs are selected from the pool and constitute the input for the next extraction sub-step. These two steps are usually of complexity that is linear in time with respect to the size of the given instance. This is the conceptual strength of DNA computing, because the pools that are elaborated in the steps of the procedure are of a size that is exponential with respect to the size of the given instance. Generation of the solution space can be performed in several manners, by using the power of DNA recombination [1, 10], or by a sequence of steps according a Mix-and-Split procedure [3]. Extraction remains the critical point of this paradigm. For example, the method which uses the biotin-avidine affinity to select some strands by means of the complementary substrands bond to beads has efficiency $88 \pm 3\%$. Moreover, in the context of the Adleman-Lipton extract model, as it is reported in [7], if we call p the percentage of sound extractions (extracting each of the required DNA strands is equally likely) and we assume that for each distinct string s in a test tube there are 10^l ($l = 13$ proposed by Adleman in [1]) copies of s, then no matter how large l is and no matter how close to 1 p is, there always exists a class of 3-SAT problems such that DNA computing error must occur.

C. Ferretti, G. Mauri and C. Zandron (Eds.): DNA10, LNCS 3384, pp. 104–112, 2005.

In this paper we address the following particular problem. Given a specified sequence γ of bases, and an input pool P of different dsDNA molecules with a same length n and sharing a common prefix and suffix, we want to produce an output pool P' where only the strands which include the given sequence γ are represented. We will show that by using PCR in a particular manner, combined with gel-electrophoresis, we are able to solve this problem.

PCR is one of the most important and efficient tool in biotechnological manipulation and analysis of DNA molecules. The main ingredients of this reaction are polymerase enzymes which implement a very simple and efficient duplication algorithm on double oriented strings. The PCR procedure is based on: i) templates, ii) a copy rule applied to templates, iii) initial short strings (primers) that say where the copying process has to start. Polymerase enzyme 'writes' step by step, in the $5' - 3'$ direction, by coping (in complementary form) the bases of the template which drives the copy process. The bilinearity of DNA molecules and the antiparallel orientation of their two linear components are essential aspects of the logic underlying the whole process [5, 11].

The idea of using PCR as a "very elegant and effective detection method" to check the existence of a solution was considered in [12], where a theoretic method 'blocking' the wrong sequences with PNA strands is proposed, and a paper on experimental aspects of DNA computing by blocking was announced.

In the next section we propose the XPCR procedure, that is a variant of standard PCR where two dsDNA molecules and two primers are used in such a way that one primer hybridizes with one single strand of a DNA molecule, and the other primer with one single strand of the other DNA molecule. A similar idea, but in a very different context, was considered in [9]. In section 3 we present an extraction algorithm based on XPCR, and we give some experimental results that show its validity.

2 Cross Pairing PCR

Firstly we introduce some terminology and notation.

As usual, we intend that upper strands are in the direction from left to the right ($5' - 3'$) and lower strands are in the opposite direction. We use the terms 'string' and 'sequence' in an almost equivalent manner, however, we speak of *strings* when we want to stress their abstract concatenation structure as words of a free monoid over a finite alphabet (the Watson Crick alphabet in our specific case). We adopt Greek letters for strings, and $\bar{\alpha}$ will indicate the reversed complementary string of α. *Strands* are physical (oriented and rotational, single or double) DNA instances of strings. We say γ-strand any strand (single or not) which is an instance of string γ, and γ-superstrand any strand which includes a γ-strand. Moreover, shortly we say that $\alpha \in P$ when α-strands are present in the pool P.

In the following we will explain in an informal manner the steps of the extraction algorithm we propose in this paper, by using as much as possible a pictorial language, but it is interesting to note that by extending the bilinear notation introduced in [5, 11] we could express all the process we consider in a complete formal way.

If one puts in a test tube many copies of $\alpha \ldots \gamma$-strands and $\gamma \ldots \beta$-strands (that finish and start respectively with a same substring γ), and two primers α and $\bar{\beta}$, then PCR performs the process one can see in figure 1, where $\alpha \ldots \gamma \ldots \beta$-

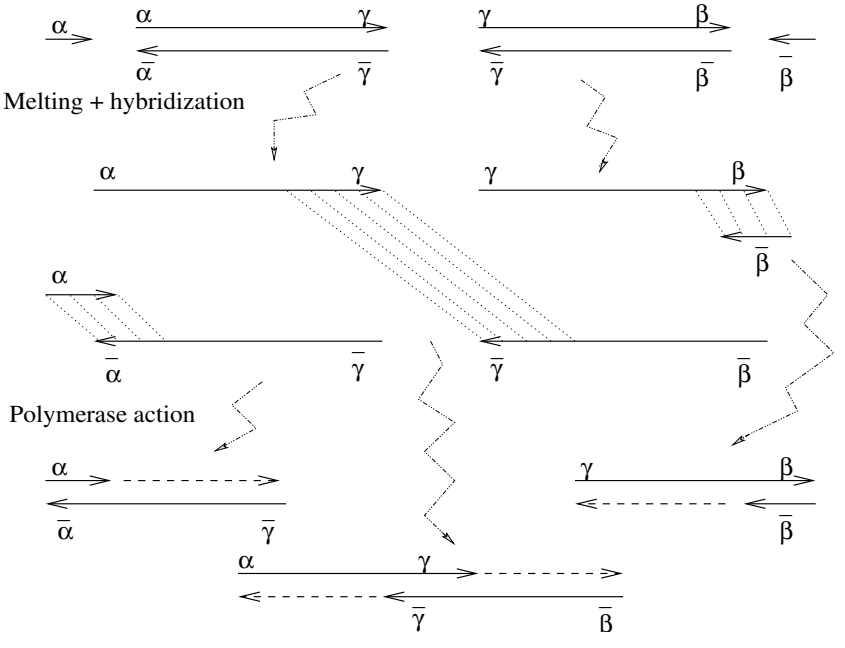

Fig. 1. Basic step of XPCR

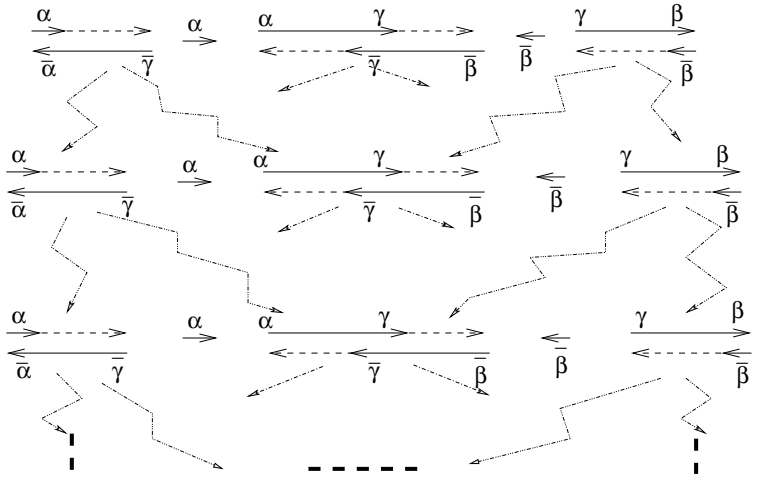

Fig. 2. Amplification of XPCR. Long strands in the middle are seeds of exponential amplifications

strands are firstly generated, by a sort of overlapping juxtaposition of the two initial strands, and then amplified. This step has been verified in laboratory in different situations and variants, with three sizes of γ (229, 95, 15 bp, see footnotes 1, 4, 6 in section 3.1).

As one can see in figure 1, differently from standard PCR the primers hybridize with single strands of two different dsDNA molecules, so liberating the respective partners in each molecule. At this point, these single strands can hybridize each other by means of their (reversed) complementary parts γ and $\bar{\gamma}$, and the polymerase uses the single strand components of this structure as templates in order to complete the double string. This process is the key point of the extraction algorithm of the next section (see step 4). Note that only the long strings are amplified in each step (see figure 2); in fact, one of the strands of the initial short pieces in every step is the template for the generation of another short piece, and the two generated short pieces will join to form a long string including γ.

3 Extraction Algorithm

Let us start with a pool P which is constituted by dsDNA molecules having same length n, α-strands at beginning and β-strands at end. Given a string γ, let us assume that P is γ-*invariant*, that is, either γ does not occur at the same position in different strands of P, or if it is not the case, then $\alpha\tau_1\gamma\tau_2\beta, \alpha\tau_3\gamma\tau_4\beta \in P$ implies that $\alpha\tau_1\gamma\tau_4\beta, \alpha\tau_3\gamma\tau_2\beta \in P$. The hypothesis of a common prefix and suffix and the γ-*invariance* of the pool are not restrictive assumptions in the context of DNA computing. The following procedure gives as output a pool P' where all the γ-superstrands of P are represented.

In the pictures related to each step of the algorithm we do not mention the products given by secondary linear amplification, because ignoring them will not affect the validity of the procedure. We use the notation PCR(α, β) to indicate a PCR performed with α as forward primer and β as reverse primer.

1. **PCR**$(\alpha, \bar{\gamma})$
   After this step we find in the test tube an exponential amplification of the dsDNA $\alpha \ldots \gamma$ (see figure 3) that are shorter than the initial molecules (products linearly amplified keep the initial length).
2. **PCR** $(\gamma, \bar{\beta})$
   After this step we find in the test tube an exponential amplification of the dsDNA $\gamma \ldots \beta$ (see figure 4) that are shorter than the initial molecules (products linearly amplified keep the initial length).
3. **Gel-electrophoresis for selecting the short strands of lengths** l_1, l_2 **such that** $l_1 + l_2 - l = n$ **where** l **is the length of** γ
   In this step only the strings $\alpha \ldots \gamma$ and $\gamma \ldots \beta$ are selected.
4. **XPCR**$(\alpha, \bar{\beta})$
   In this step all the γ-superstrands of P, that are the longest ones in the current pool (see figure 5), are exponentially amplified.
5. **Gel-electrophoresis for selecting the** n**-long strands**.

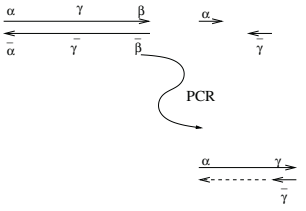

Fig. 3. First step based on PCR

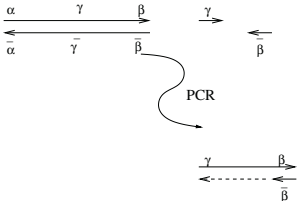

Fig. 4. Second step based on PCR

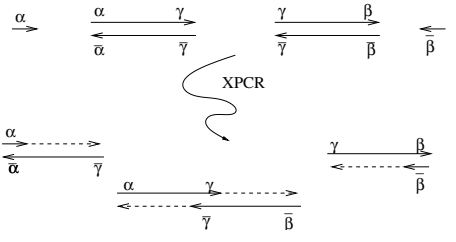

Fig. 5. Fourth step based on XPCR

3.1 Encoding and Experiments

A very important aspect of our method is the type of encoding we need in order to avoid unexpected phenomena of annealing. In fact, primers have to work in a very specific manner. We adopted a comma free encoding, following some of the general principles given in [2, 4], by using a program that checked the strings of the pool and primers according to the following strategy. A test $T(n, m)$ is positive when a situation is found such that in a window of n consecutive positions there are at least m discordance positions This means that when $T(n, m + 1)$ is negative then a window there exists with more than $n - m + 1$ concordance positions (in this case the program localizes and shows all these windows). Several tests were performed with different values of the parameters n and m related to the size of our primers.

An important *caveat* is the 'primer rotation' phenomenon. When it occurs, a forward primer can play the role of another reverse primer or *vice-versa*. We

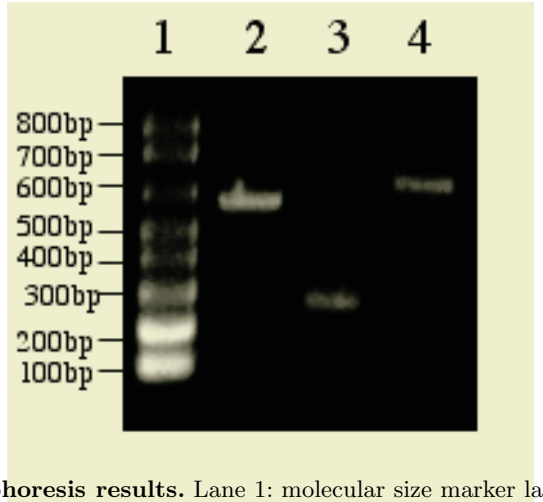

Fig. 6. Electrophoresis results. Lane 1: molecular size marker ladder (100b). Lane 2: $\alpha\phi\gamma\psi$-strands of human RhoA (582bp), lane 3: $\gamma\psi\beta$-strands (253bp), lane 4: cross pairing amplification of $\alpha\phi\gamma\psi\beta$-strands (606bp): 606 = 582 +253 - 229

collected the outcomes of many experimental trials that suggested us the values of parameters that ensure a reliable behavior of the primers. However, oligos and primers we used in the experiment do not have common strings of 5 bases long (apart the expected annealing part, where the concordance is total in a window with the same length of the primer).

In order to test the validity of XPCR, a first experiment was carried on with $\alpha\phi\gamma\psi$-strands, $\gamma\psi\beta$-strands, and primers α and $\bar{\beta}$ (data are shown in figure 6)[1], where $\alpha\phi\gamma\psi$ was RhoA human gene which regulates many essential cellular processes and controls cell activation in response to environmental cues. In the following $-n$ position of a sequence indicates the position n in the backward direction.

Other experiments were performed with pools of sequences $\alpha\tau_2 \dots \tau_9\beta$ (150 long) generated by a combination of strands $X_2, X_3, \dots, X_9$, Y_3, Y_4, ..., Y_8, Z_2, Z_4, Z_6, Z_7, Z_9 [2] in such a way that τ_i is equal to X_i or Y_i or Z_i (for each i we have at least two choices). In particular, we started from a pool of eight different

[1] α = ATGGCTGCCATCCGGAAG, γ = GAAGGATCTTCGGAATGATG ... at position -229 of Rho A, and $\bar{\beta}$ = GAACAGAAACTTATCTCAGAGGAA.

[2] X_2 = CAAGATATGG, X_3= TCGTCTGCTAGCATG, X_4 = TCACGCCACGGAACG, X_5 = GTGAGCGCGAGTGTG, X_6 = ATATGCAATGATCTG, X_7 = ATCCGTCCCGATAAG, X_8 = CAAGTCAGATTGACC, X_9 = GCACGTAACT, Y_3 = CCCGATTAGTACAGC, Y_4 = TACTGATAAGTTCCG, Y_5 = TCGCTCCGACACCTA, Y_6 = TCAGCCGGCTTGCAC, Y_7 = AACTGATACGACTCG, Y_8 = TATTGTCACGCATCG, Z_2 = CAAGAGATGG, Z_4 = TCACGCCACGGAACT, Z_6 = TTAGCCGGCTTGCAC, Z_7 = TACTGATACGACTCG, Z_9 = GTACGTAACT, α = GCAGTCGAAGCTGTTGATGC, β = AGACGCTGCCGTAGTCGACG.

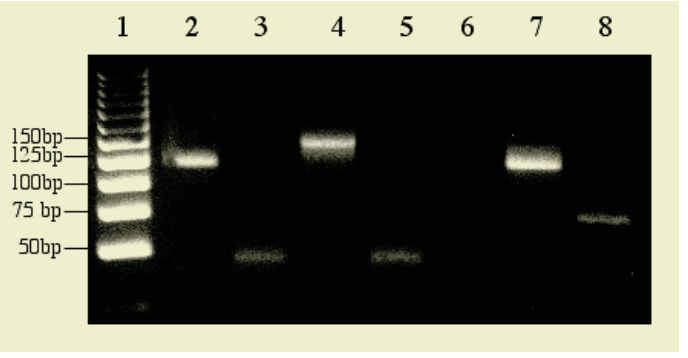

Fig. 7. Electrophoresis results. Lane 1: molecular size marker ladder (25 bp). Lane 2: amplification of $\alpha \dots \gamma$ strands (120 bp); lane 3: amplification of $\gamma \dots \beta$ strands (45 bp); lane 4: cross pairing amplification of $\alpha \dots \gamma$ and $\gamma \dots \beta$ (150 bp). Lane 5: positive control by PCR$(\gamma, \bar{\beta})$, with γ at position -45; lane 6: negative control by PCR$(\gamma', \bar{\beta})$; lane 7, 8: positive controls by PCR$(\gamma_1, \bar{\beta})$ and PCR$(\gamma_2, \bar{\beta})$ respectively, with γ_1 at position -125 and γ_2 at position -75

types of strands[3] where γ-superstrands $\alpha \dots \gamma$ and $\gamma \dots \beta\delta$ were obtained by standard PCRs and again also in this case the expected results: $\alpha \dots \gamma \dots \beta\delta$-strands were produced by XPCR [4].

Finally, the complete algorithm was tested on a pool[5] in which γ' is present only in all the sequences that are not γ-superstrands[6] and all the γ-superstrands are either γ_1-superstrands or γ_2-superstrands[7]; all the steps were proved to be correctly performed (see lanes 2, 3, 4, 5 of figure 7), in fact all and only γ-superstrands were extracted from the pool. In order to verify the correctness of our results we performed three PCRs on the final test tube. The first one with primers γ' and $\bar{\beta}$ showed that only γ-superstrands were present in the final pool, because there was no amplification of γ'-superstrands (see lane 6 of figure 7). The last two PCRs with primers γ_1, $\bar{\beta}$, and γ_2, $\bar{\beta}$, respectively, showed that all the initial γ-superstrands were present in the final tube (see lanes 7, 8 of figure 7).

[3] $\alpha\ Z_2\ X_3\ X_4\ X_5\ X_6\ X_7\ Y_8\ Z_9\ \beta,\ \alpha\ X_2\ Y_3\ X_4\ X_5\ Z_6\ Y_7\ Y_8\ Z_9\ \beta,\ \alpha\ X_2\ Y_3\ X_4\ X_5\ X_6\ Z_7\ X_8\ X_9\ \beta,\ \alpha\ Z_2\ X_3\ Y_4\ Y_5\ Y_6\ Y_7\ X_8\ X_9\ \beta,\ \alpha\ Z_2\ X_3\ Z_4\ Y_5\ Y_6\ X_7\ Y_8\ Z_9\ \beta,\ \alpha\ X_2\ Y_3\ Y_4\ Y_5\ Y_6\ X_7\ X_8\ X_9\ \beta,\ \alpha\ Z_2\ X_3\ Y_4\ Y_5\ Y_6\ X_7\ Y_8\ Z_9\ \beta,\ \alpha\ X_2\ Y_3\ Y_4\ Y_5\ Y_6\ Y_7\ X_8\ X_9\ \beta$.

[4] γ = GAACGGTGAGCGCGAGTGTG ... in position -95 in all the strands where it occurs, δ = CTTGTCTTTGAATAGAGTCTCCTT.

[5] $\alpha\ Z_2\ X_3\ X_4\ X_5\ X_6\ X_7\ Y_8\ Z_9\ \beta,\ \alpha\ X_2\ Y_3\ X_4\ X_5\ Z_6\ Y_7\ Y_8\ Z_9\ \beta,\ \alpha\ Z_2\ X_3\ Y_4\ Y_5\ Y_6\ Y_7\ X_8\ X_9\ \beta,\ \alpha\ X_2\ Y_3\ Y_4\ Y_5\ Y_6\ X_7\ X_8\ X_9\ \beta,\ \alpha\ X_2\ Y_3\ Y_4\ Y_5\ Y_6\ Y_7\ X_8\ X_9\ \beta$.

[6] $\gamma = Y_8$, $\gamma' = Y_4$.

[7] γ_1 = GATGGTCGTCTGCTAGCATG , γ_2 = TTAGCCGGCTTGCAAACTG.

4 Conclusions

In general and abstract terms XPCR is a method for performing, in a cheap and efficient manner, the following transformation on strings that is essentially the splicing combinatorial mechanism in the sense of the original formulation introduced by Tom Head in [6] (more precisely, a case of *null context splicing* rule):

$$\alpha\,\phi\,\gamma\,\psi\,\beta, \quad \alpha\,\delta\,\gamma\,\eta\,\beta \qquad \longrightarrow \qquad \alpha\,\phi\,\gamma\,\eta\,\beta, \quad \alpha\,\delta\,\gamma\,\psi\,\beta.$$

We showed that this method is useful for selecting γ-superstrands from a given pool, but we think that XPCR could have also a more general relevance in the context of DNA computing. of solution spaces of In principle, consecutive DNA extraction from a given pool, by means of XPCR, should work correctly, but problems could arise if the encoding is not robust enough for avoiding unexpected annealing. Therefore, we intend to develop, in the next future, encoding methods that make iterated XPCR reliable. Applications and extensions of cross pairing amplification and of the extraction algorithm based on it will also be topics of future researches.

5 Appendix (Experimental Protocols)

Reagents. 25 bp and 1 kb marker DNA ladder (Promega); agarose (Gibco brl); PCR buffer, $MgCl_2$ and dNTP (Roche); Taq DNA Polymerase (produced in laboratory); all the synthetic DNA oligonucleotides 150 bases long and all the primers were from Primm s.r.l. (Milano, Italy).

Annealing of Synthetic DNA Oligonucleotides. Two complementary synthetic 150 bases long DNA oligonucleotides ($5' - 3'$ and $3' - 5'$) were incubated at 1:1 molar ratio at 90° for 4 min in presence of 2.5 mM of $MgCl_2$ and then at 70° for 10 min. The annealed oligos were slowly cooled to 37°, then further cooled to 4° until needed.

PCR Amplification. PCR amplification was performed on a PE Applied Biosystems GeneAmp PCR System 9700 (Perkin Elmer, Foster City, CA) in a 50μl final reaction volume containing 1.25U of Taq DNA Polymerase, 1.5 mM $MgCl_2$, 200 μM each dNTP, PCR buffer, 50 ng DNA template, 0.5-1 μM of forward and reverse primers. The reaction mixture was preheated to 95° for 5 min. (initial denaturing), termocycled 25 times: 95° for 1 min. (denaturing), 58° for 1 min. (annealing), 72° for 15 sec (elongation); final extension was performed at 72° for 10 min.

Preparation and Running of Gels. Gels were prepared in 7x7 cm or 6x10 cm plastic gel cassettes with appropriate combs for well formation. Approximately 20 or 35 ml of 4% agarose solutions were poured into the cassettes and allowed to polymerize for 10 min. Agarose gels were put in the electrophoresis chamber and electrophoresis was carried out at 10 volt/cm^2, then the bands of the gels

were detected by a gel scanner. The DNA bands (PCR products) of interest were excised from the gel and the DNA was purified from the gel slices by Promega Kit (Wizard SV Gel and PCR Clean-Up System).

Acknowledgments

We kindly thank Prof. Angelo Spena of the Biotechnological Department of the University of Verona for his interesting suggestions on PCR phenomenon, and Luca Bianco of the Department of Computer Science of the University of Verona for his collaboration in the encoding of sequences.

References

1. Adleman L. M., *Molecular Computation of solutions to combinatorial problems*, Science **266**, pp 1021-1024, 1994.
2. Arita M., *Writing Information into DNA*, in: Aspects of Molecular Computing, N. Jonoska, G. Păun, G. Rozenberg (eds.), LNCS 2950, pp. 23-35, 2004.
3. R.S. Braich, N. Chelyapov, C. Johnson, P.W.K. Rothemund, L. Adleman, *Solution of a 20-Variable 3-SAT Problem on a DNA Computer*, Science **296**, pp 417-604, 2002.
4. R.S. Braich, C. Johnson, P.W.K. Rothemund, D. Hwang, N. Chelyapov, L. Adleman, *Solution of a Satisfiability Problem on a Gel-Based DNA Computer*, A. Condon, G. Rozenberg (eds): DNA Computing, 6th International Workshop on DNA-Based Computers, Leiden, Netherlands. Lecture Notes in Computer Science, Springer-Verlag **2054**, pp 27-42, 2000.
5. G. Franco, V. Manca, *An algorithmic analysis of DNA structure*, Soft Computing, to appear, 2004.
6. T. Head, *Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviors*, Bulletin of Mathematical Biology **49**, pp 737-759, 1987.
7. D. Li, *Is DNA computing viable for 3-SAT problems?*, TCS **290**, pp 2095-2107, 2003.
8. R.J. Lipton, *DNA solutions of hard computational problems*, Science **268**, pp 542-544, 1995.
9. S. Liu, D. S. Thaler, A. Libchaber, *Signal and noise in bridging PCR*, BMC Biotechnology, **2**:13, 2002.
10. V. Manca, C. Zandron, *A Clause String DNA Algorithm for SAT*, N. Jonoska, N.C. Seeman (eds.): DNA Computing, 7th International Workshop on DNA-Based Computers, Tampa, Florida. Lecture Notes in Computer Science **2340**, Springer, pp 172-181, 2001.
11. V. Manca, *On the logic of DNA bilinearity*, Preliminary Proceedings, Masami Hagiya, Azuma Ohuchi (Eds.): DNA Computing, 8th International Meeting on DNA Based Computers, Hokkaido, Japan, p 330, 2002.
12. G. Rozenberg, H. Spaink, *DNA computing by blocking*, TCS **292**, pp 653-665, 2003.