A Molecular Algorithm Solving NP-Complete Problems by Mimicking the Cell

Technical Report

Ali A Atiia
School of Computer Science
McGill University

2015

1 Executive Summary

Molecular computing is a new perspective, not a technological invention. Research in the field is conducted in reverse to the traditional relationship between biology and computer science where biological objects and relations are simulated *in silico*, under the banner of bioinformatics. The field was born as a result of an new way of attributing real-world objects/relations to biomolecules, as they have come to represent, say: nodes/edges in a graph [1], variables/truth values in a SAT formula [2], or tiles/tiling in a recreational edge-matching puzzle [3]. There is no structural, chemical or enzymatic operation used in molecular computing that does not already occur in nature, and the manipulation of biomolecules follows standard procedures that are routinely carried out in molecular biology laboratories. The novelty therefore is neither in materials nor in methods, but rather in perspective. DNA/RNA/protein sequence compositions and the chemical and enzymatic reactions that govern their interactions are given new *semantic* interpretations that reflect computational problems and algorithms. Presented here is a proposal for the design and implementation of a molecular algorithm to solve an instance of the NP-complete Hamiltonian path problem (HPP) by mimicking the functioning of the cell. The presented molecular algorithm has inspired a simple graph model for how genetic regulatory networks evolve and function in general, susequently establishing a link between computational intractability and the evolution of biological networks into their current topology [4].

2 Background

At the intersection between biology and computer science lie two related areas of scientific inquiry. In one direction, *in silico* models of biological components/processes are used to gain insight into biology through algorithmics as exemplified in the field of bioinformatics. On the opposite direction, and since Adleman's insightful paper [1], computational problems (algorithms) are modeled (implemented) using biological components (processes) in a line of research that has rapidly evolved into the field of molecular computing.

Adleman's proof-of-concept demonstration [1] involved the encoding of nodes and directed edges of a small 7-node graph with DNA strands. The interactions between strands and the enzymatic operations carried out subsequently constituted a molecular algorithm that resulted in a long double-stranded DNA encoding for an answer to the question: does there exist a path from Node 0 to Node 6 that visits every other node along the way *exactly* once (i.e. a Hamiltonian path between Node 0 and Node 6). Figure 1a demonstrates Adleman's scheme on just two nodes labeled Montreal and Toronto each encoded with a 16-nucleotide (nt) single-stranded DNA (ssDNA). The directed edge between the two cities (representing a highway for example) is also a 16-nt ssDNA that is Watson-Crick complementary to Montreal's strand in part and to Toronto's in another. Figure 1b shows the basic molecular procedure that follows. The mixing of the three strands under standard buffer conditions results in hydrogen bond formation according to A-T and G-C complementary, and the statement "there exists a path from Montreal to Toronto" is made permanent by the ligation (covalent concatenation) of the two strands using a ligase enzyme. Such concatenation product can selectively and exponentially be amplified using a polymerase enzyme (of the same enzyme family as those amplifying a genome in a dividing cell) thru a polymerase chain reaction (PCR) resulting in a double-stranded DNA (dsDNA). If one began with all sequences representing all nodes and edges in a graph, and a large enough quantity of each strand is present in the

mix (indeed, here's where computational intractability strikes), then the post ligation product constitutes a brute force search over all possible paths in the graph. Standard molecular techniques are subsequently applied in order to 'fish-out' the dsDNA sequence that constitutes the concatenation of nodes encoding the correct Hamiltonian path.

Despite the massive parallelism of molecular computations (up to 10^{17} parallel ligations can take place in a single tube, using micromole amounts of strands[1]), that parallelism is dwarfed by the exponential resource consumption that *any* algorithm for solving NP-complete problems requires on worst-case instances (provided P ≠ NP), and so computational intractability still manifests itself in the exponential molarity of each species. Indeed, shortly after Adleman's demonstration, Hartmanis showed how an HPP of a 200-city tour solved using Adleman's method (which is brute-force) would require an amount of DNA that is more the weight of the Earth [5]. Previously we analyzed upper-bound molarity requirement on a solution to an instance of the NP-complete Edge-Matching Puzzle [3], and experimentally demonstrated how even when the powerful polymerase chain reaction (PCR) procedure is used to concentrate strands prior to the brute-force step, the number of PCR cycles would still grow exponentially as the problem size grows linearly.

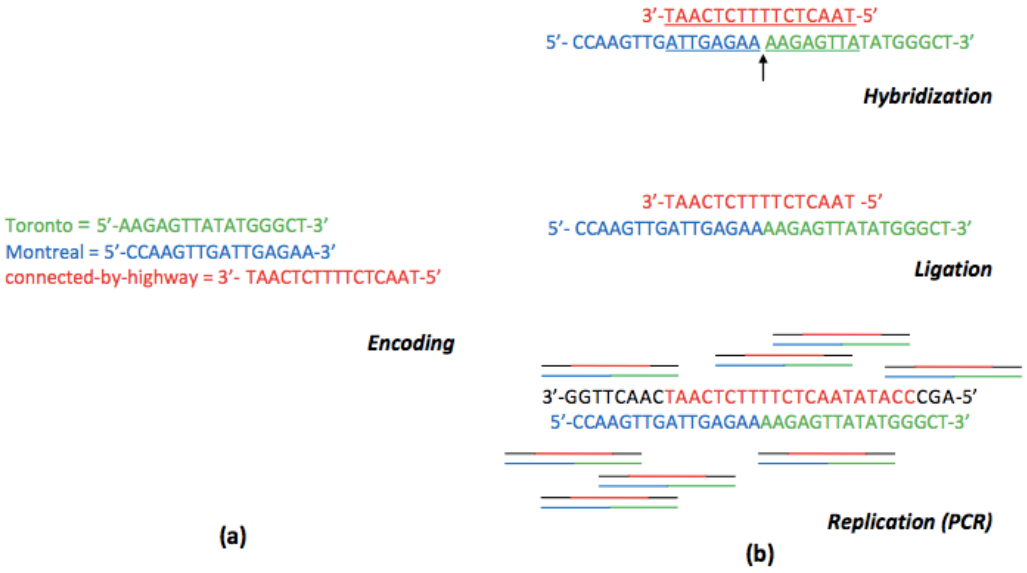


Figure 1: Demonstrating Adleman's HPP solving algorithm. **(a)** Encoding of two nodes (cities) and an edge (the relation "connected-by-highway") using 16-nt ssDNA sequences; the edge sequence is partially Watson-Crick (WC) complementary to the each node. **(b)** The three fundamental operations of Adleman's molecular algorithm for solving HPPs. The WC-complementarity brings two strands together by hydrogen-bond formation under standard buffer conditions, with a "nick" at their meeting point (indicated by an arrow), which is

subsequently sealed by a ligase enzyme. A strand of interest in the ligation mix can be selectively & exponentially amplified by PCR, resulting in fully double-stranded DNA (dsDNA).

Nonetheless, Adleman's demonstration generated considerable interest and inspired a wide range of new research proposing and implementing various molecular algorithms [6]. Moreover, many computer scientists have since then been inspired to use DNA in the material world, ranging from Turing-universal DNA models [7], to DNA finite state machines for control of gene expressions (or what can be referred to as *in vivo* molecular automata [8][9]), to DNA nanotechnological constructions [10][11].

3 Method

3.1 Overview

The proposed molecular algorithm for solving Adleman's instance of the Hamiltonian path problem (A-HPP) follows the working of the cell: encoding the problem with DNA, executing the algorithm as competitive RNA ligation, and printing the result as a protein. The DNA → RNA → protein information processing mechanism is what is refered to as the "central dogma" of molecular biology [12], and is the general information processing mechanism in almost all living organisms. Since HPP is NP-complete, all problems in NP are polynomial-time reducible to it by Cook-Levin theorem [13] and so, from a computational perspective, the algorithm demonstrates that the cell can, in principle, decide all languages in NP. Various contrived models of computing with biomolecules [14], as well as models based on existing biological systems [15][16] have been shown to Turing-universal. There is, however, a fundamental logical limitation to attempting to de-complexify biological systems in general (say, a human cell) since, by Church-Turing thesis, there is a Turing machine T that can simulate the transcription and translation actions through a cell's life time but, by Rice's theorem [17], it is undecidable to deduce any non-trivial property of T.

3.2 Experimental approach

The presented molecular algorithm to solve A-HPP instance (Figure 2a) mimics the information processing mechanism in living organisms. The original Adleman's algorithm [1] is as follows:

Step 1: Generate random paths through the graph

Step 2: Keep only those paths that begin with v_{in} [N_0] and end with v_{out} [N_6]

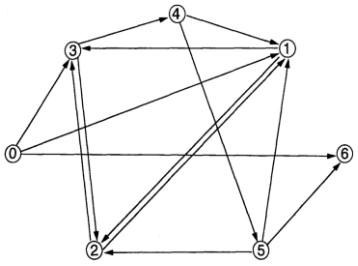
Step 3: If the graph has n nodes, keep only those paths that enter exactly n nodes

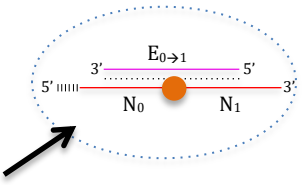
Step 4: Keep only those paths that enter all of the nodes of the graph at least once

Step 5: If any paths remain, say "Yes"; otherwise say "No"

We first describe the experimental procedure to implementing these steps in general terms (a more detailed exposition to follow in section 3.3). Figure 2b shows a schematic representation of the encoding of A-HPP with DNA, the brute-force computation of the solution as a competitive RNA ligation reaction, and the solution printout as a protein. To implement Step 1, we assign to each node and each edge a double-stranded DNA (dsDNA) sequence primed upstream with a T7 promoter region. The dsDNA sequences subsequently serve as templates in an *in-vitro* transcription reaction. An RNA transcript of edge $E_{i\to j}$ is, by design, partially complementary to N_i at the 3′ end and partially complementary to N_j at the 5′ end. The RNA transcripts are pooled and splint-ligated (edges = splints) generating random paths (except for N_0, transcription of all nodes are primed with Guanosine monophosphate (GMP) to facilitate ligation, more details in section 3.3 step (3)). The fact that hybridizing strands must be in opposite 5′-3′ orientation conveniently translates into and preserves edge directionality in A-HPP graph, and so edge 3′-[$E_{0\to1}$]-5′ hybridizes always to 5′-[N_0]-3′-5′-[N_1] but never to 5′-[N_1]-3′-5′-[N_0], as shown in Figure 2b (top inset). RNA transcripts are ligated at high concentration of each species, ensuring the generation of each partial path, correct or otherwise, with high probability (see [1] for a details on the minimum amount stochastically required).

a)





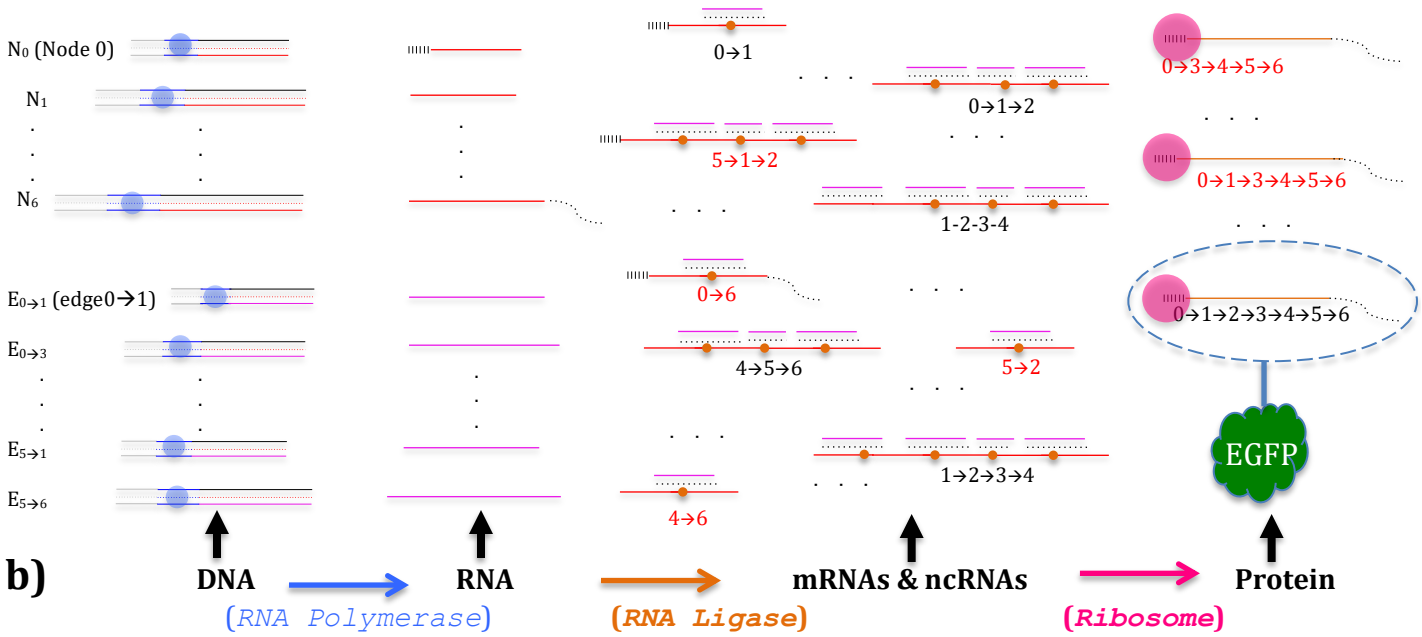


Figure 2: Simulation of a solution to Adleman's HPP as a cellular process. A) Adleman's HPP (A-HPP) graph. B) Schematic representation encoding and solving A-HPP; left to right: dsDNA templates for each node & edge in A-HPP, each template primed with leading (grey) sequence and T7 promoter (blue) sequence. T7 polymerase (blue circle) transcribes an RNA strand from each DNA template (orange lines for nodes; purple for edges), N_0's sequence is primed with m7G analog (NEB) and contains a leading ribosome-binding site [18] (black vertical bars), N_6's transcript is polyadenylated (curvy dotted line) with *E. coli* poly(A) (NEB). RNA transcripts are ligated with T4 RNA ligase 2 (NEB) (orange circles) leading a brute-force exhaustive generation of correct and erroneous paths (black & red subscripts, respectively). Ligated sequences beginning with N_0 and ending with N_6 represent mRNAs suitable for translation by the ribosome (by virtue of m7G cap & RBS for translation initiation in N_0, and poly(a) tail for mRNA stability in N_6), but only the correct path encodes the EGFP protein (by deliberate design); since nodes' sequences have different lengths, the correct solution's protein has a unique kD weight; correctness of node sequence in the solution path is validated by the fluorescence of EGFP.

To implement Step 2, the transcription reaction of N_0 is primed with m7G analog (NEB), and its sequence contains a ribosome-binding site (RBS) at the 5′ end [18]; while N_6 sequence is polyadenylated using *E. coli* poly(A) polymerase. These properties of N_0 and N_6 ensure that only RNA ligation products beginning with N_0 and ending with N_6 have ribosomal affinity (m7G and RBS) and transcript stability (poly(A)) rendering them suitable for translation into proteins.

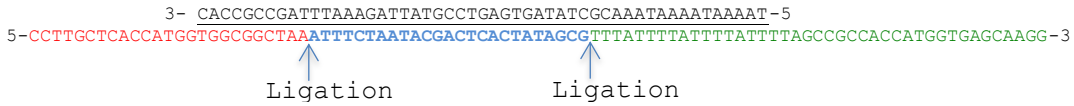
To implement Step 3-5, Adleman's procedure can be followed (exclusion by gel excision, followed by magnetic bead immobilization). However, these three steps can be implemented in one shot if careful sequence design is followed. The concatenated (ligated) sequence $N_0 \rightarrow N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_4 \rightarrow N_5 \rightarrow N_6$ is (by our design) the mRNA sequence of EGFP protein (enhanced green fluorescent

protein [19]). The kilo Dalton (kD) weight of EGFP combined with its unique fluorescence property eliminates the need for selection by length (Step 3) and uniqueness (Step 4). Ligation products that begin with N_0 and N_6 but do not contain all the other nodes (or contain all of them with repetitions) will translate into proteins with different kD weight. Ligation products that begin with N_0 and end with N_6, and contain all the other nodes (N_1 , N_2 , N_3 , N_4 , N_5) exactly once but in an erroneous order, such mRNAs will not fluoresce because the resulting mRNA sequence is drastically different from the correct solution even with one sequence misplaced. Hence, only the correct solution will result in EGFP and its existence can readily be verified under fluorescence microscopy.

3.3 Anatomy of DNA templates, RNA transcripts, and reaction conditions for a proof-of-concept demonstration:

This section presents a more detailed view of the DNA templates used to encode nodes and edges, the resulting RNA transcripts which are the participants in the competitive RNA ligation, and the reaction conditions in each step of the experimental procedure:

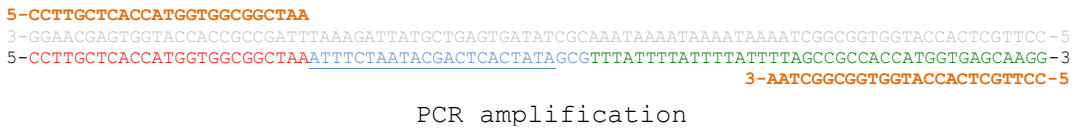
(1) DNA templates were ordered as dsDNA blocks (IDT gblocks) or constructed using ligation from smaller synthesized oligonucleotides (Biocorp) as follows. Segments of each node/edge sequence were concatenated using splint ligation with T4 DNA ligase. The following example illustrates the assembly and ligation of N_0's DNA template:



The splint oligonucleotide (black, underlined) and the constituent oligonucleotides of a node (red, blue, and green) were mixed at 10 uM concentration each in a 30-ul ligation reaction (50 mM Tris-HCl, 10 mM MgCl2, 1 mM ATP, 10 mM DTT. 13u/ul T4 DNA ligase (NEB)) for 1 hour at room temperature. Donor oligonucleotides (blue and green) are phosphorylated with T4 Polynucleotide Kinase (PNK) (NEB) prior to ligation, and in the same ligation reaction conditions. PNK adds a phosphorous residue at the 5′ end of donor strands, a prerequisite for ligation.

(2) The ligation product is used as template in a PCR reaction. 0.5 ul of ligation reaction (5 picomole total concentration) is used as template in a 25-ul PCR reaction using Phusion DNA polymerase (NEB) and carried out for 40 cycles. By design, one oligonucleotide serves as both the forward and backward

primer (since the primer sequence, red in this example, is purposefully ligated 5′ of the template). This eliminates the need to optimize melting temperature condition to satisfy two primers. The following illustration shows the resulting dsDNA template from PCR for N_0, with primer sequence shown in bold orange, and the T7 promoter region underlined:



PCR amplification

(3) PCR products are used as template in *in-vitro* transcription (IVT) reactions at a concentration of 20 ng/ul in total reaction volume of 50ul (40 mM Tris-HCl, 6 mM MgCl2, 1.5 mM DTT, 2 mM spermidine, 1U/ul T7 (NEB)). The reaction is carried out for 2-4 hours at 37 degrees Celsius and subsequently treated with 5 units of DNAse I (NEB). Transcription reactions contained 2mM concentration of each NTP, except for N_0 where GTP was added to 0.5mM concentration while m7G analog (NEB) was added to 4 mM concentration (to facilitate ribosomal translation of transcripts beginning with N_0). In IVT reactions of N_1 to N_6, guanosine monophosphate (GMP) (Sigma) was added to a 2mM concentration while guanosine triphosphate (GTP) was added to a 0.5mM concentration (to facilitate RNA ligation, since ligase requires monophosphate at the 5′ donor RNA). The example below shows N_0's IVT, with the arrow indicating the transcription start site of T7 polymerase. In all templates, the 1st transcribed base is G (preferred by T7) and the 2nd/3rd are CG when possible, as this has been shown to further improve transcription yield [20]:

Transcription

(4) N_6 RNA sequence is polyadenylated using *E. coli*. poly(A) polymerase (NEB) in a total reaction volume of 10ul at concentration of 5 ng/ul (50 mM Tris-HCl 250 mM NaCl 10 mM MgCl2, 0.5U/ul poly(A)), in order to facilitate ribosomal translation of sequences ending with N_6 since the ribosomal translation mix to be used is from eukaryotes (Promega's Human In Vitro Translation system) and polyadenation is a prerequisite for mRNA stability and successful translation [21].

(5) The RNA transcripts are ligated using T4 RNA Ligase 2 (NEB) at a concentration of 10uM each transcript (nodes and edges) in a total reaction volume of 30ul (50 mM Tris-HCl 10 mM MgCl2 2 mM DTT, 1U/ul T4 Ligase).

(6) The solution to A-HPP is an mRNA sequence encoding for the enhanced fluorescent green protein (EGFP). The translation step has not yet been implemented. The anatomy of the ligation product encoding for the correct A-HPP solution is shown below (consecutive node sequences shown in different colors, underlined sequence = ribosome binding site (RBS); AUG = start codon, which is part of N_0, UAA=stop codon, which is part of N_6; lower-case sequence at the 3′ = polyadenylation of N_6):

5′m7GCGUUUAUUUUAUUUUAUUUUAGCCGCCACCAUGGUGAGCAAGG-N_1-N_2-N_3-N_4-N_5-N_6-UAAaaaaaaaaaaaaa……-**3′**

3.4 Preliminary Results:

The DNA and RNA phases of this project have been tested on a selection of nodes/edges. Presented here are ultraviolet images of gel electrophoresis showing DNA templates, RNA transcript, and RNA ligation results (reflecting the experimental implementation of Step 1-4 of Adleman's algorithm). Figure 4 (a) and (b) show double-stranded DNA (dsDNA) templates encoding nodes N_0 to N_6 and edges $E_{0\to1}$ to $E_{5\to6}$, respectively. The inset shows the reference molecular marker (ladder) used to verify that the observed length of each dsDNA template appears on the gel at the expected length. The PCR product of N_6 dsDNA template (Figure 4a, well 7 from the left) shows erroneous bands, so the correct band is gel-excised under ultraviolet visualization and purified using the crush-and-soak method [22]. The transcription of N_6 results in a clean band corresponding to the expected length of 98-nt (well 8 in Figure 4b). RNA transcripts of $E_{0\to1}$, $E_{1\to2}$, $E_{2\to3}$, $E_{3\to4}$, $E_{4\to5}$, and $E_{5\to6}$ in Figure 4b (wells 10-15) show smears as they are loaded immediately from *in-vitro* transcription reaction without purification, while purified transcripts of N_0 to N_6 are purified prior to loading (Zymo Research kit # R1019). The length of an RNA transcript is the length of its corresponding DNA template minus the T7 promoter region (25-bp) and leading sequence (25-32 bp). For example, the length of N_6's RNA transcript is 98-nt = 152 - 29 – 25 (152-bp total dsDNA template length – 29-bp (leading) – 25-bp T7 (promoter)).

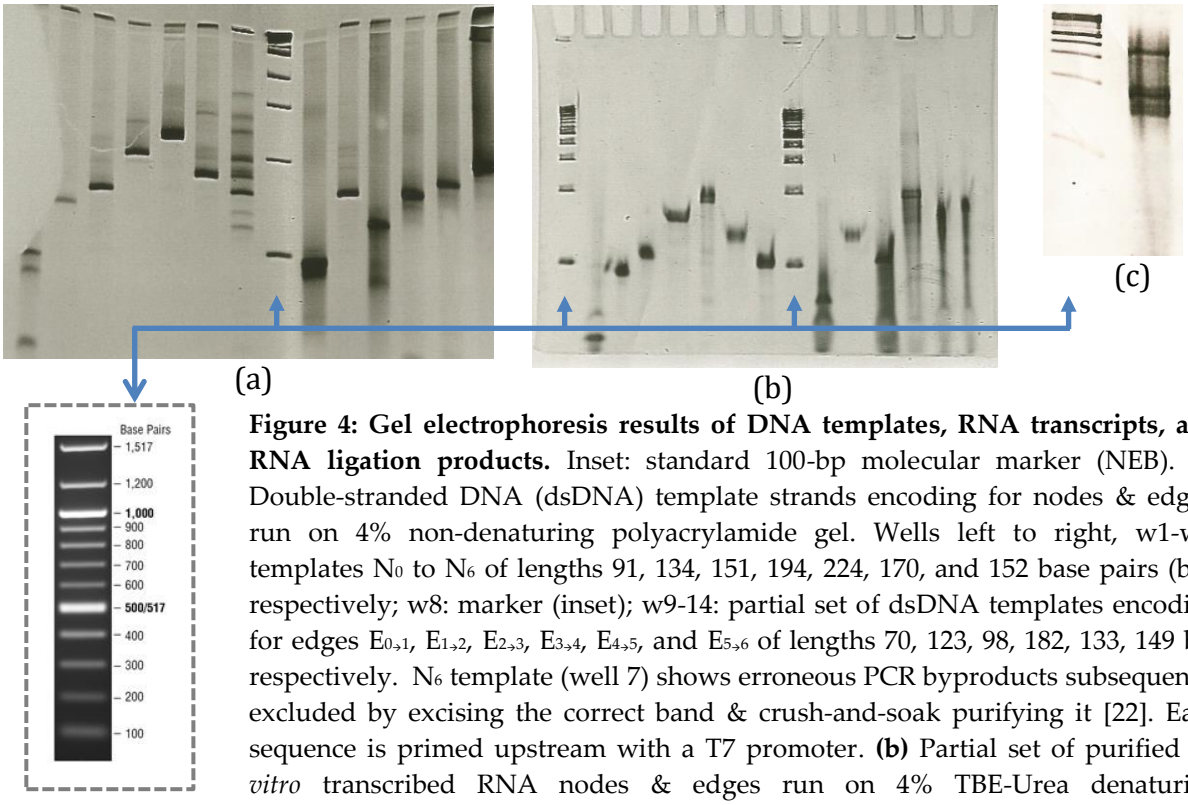


Figure 4: Gel electrophoresis results of DNA templates, RNA transcripts, and RNA ligation products. Inset: standard 100-bp molecular marker (NEB). **(a)** Double-stranded DNA (dsDNA) template strands encoding for nodes & edges, run on 4% non-denaturing polyacrylamide gel. Wells left to right, w1-w7: templates N_0 to N_6 of lengths 91, 134, 151, 194, 224, 170, and 152 base pairs (bp), respectively; w8: marker (inset); w9-14: partial set of dsDNA templates encoding for edges $E_{0\to1}$, $E_{1\to2}$, $E_{2\to3}$, $E_{3\to4}$, $E_{4\to5}$, and $E_{5\to6}$ of lengths 70, 123, 98, 182, 133, 149 bp, respectively. N_6 template (well 7) shows erroneous PCR byproducts subsequently excluded by excising the correct band & crush-and-soak purifying it [22]. Each sequence is primed upstream with a T7 promoter. **(b)** Partial set of purified *in-vitro* transcribed RNA nodes & edges run on 4% TBE-Urea denaturing polyacrylamide gel; left to right, w1/w9: marker (inset), w2-w8: RNA transcripts of N_0 to N_6 with lengths 44, 87, 102, 147, 177, 120, 98-nt, respectively; w10-w15: RNA transcripts of edges $E_{0\to1}$, $E_{1\to2}$, $E_{2\to3}$, $E_{3\to4}$, $E_{4\to5}$, and $E_{5\to6}$ with lengths 70, 123, 98, 182, 133, 149-nt, respectively. **(c)** Example RNA ligation; w1: marker (inset); w2: ligation of N_3 RNA transcript (147-nt) to N_4 (177-nt) by splint ligation with edge 3→4 transcript (182-nt) to form a 324-nt RNA strand (ligation product).

References:

[1] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5187, pp. 1021–1023, 1994.

[2] R. S. Braich, N. Chelyapov, C. Johnson, P. W. Rothemund, and L. Adleman, "Solution of a 20-variable 3-SAT problem on a DNA computer," *Science*, vol. 296, no. 5567, pp. 499–502, 2002.

[3] M. Alshamrani, "DNA Computation of Solutions to Edge-Matching Puzzles," masters, Concordia University, 2011.

[4] M. Shamrani, J. Waldispühl, and F. Major, "Evolution by Computational Selection," *ArXiv Prepr. ArXiv150502348*, 2015.

[5] J. Hartmanis, "On the weight of computations," *EATCS Bull.*, vol. 55, pp. 136–138, 1995.

[6] M. Amos, "Theoretical and experimental DNA computation," *Bull Eur. Assoc Theor Comput. Sci*, vol. 67, pp. 125–138, 1999.

[7] E. Winfree, "Algorithmic self-assembly of DNA," California Institute of Technology, 1998.

[8] Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, and E. Shapiro, "An autonomous molecular computer for logical control of gene expression," *Nature*, vol. 429, no. 6990, pp. 423–429, 2004.
[9] Z. Xie, L. Wroblewska, L. Prochazka, R. Weiss, and Y. Benenson, "Multi-input RNAi-based logic circuit for identification of specific cancer cells," *Science*, vol. 333, no. 6047, pp. 1307–1311, 2011.
[10] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman, "Design and self-assembly of two-dimensional DNA crystals," *Nature*, vol. 394, no. 6693, pp. 539–544, 1998.
[11] P. W. K. Rothemund, "Folding DNA to create nanoscale shapes and patterns," *Nature*, vol. 440, no. 7082, pp. 297–302, Mar. 2006.
[12] F. Crick and others, "Central dogma of molecular biology," *Nature*, vol. 227, no. 5258, pp. 561–563, 1970.
[13] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the third annual ACM symposium on Theory of computing*, 1971, pp. 151–158.
[14] L. Qian, D. Soloveichik, and E. Winfree, "Efficient Turing-universal computation with DNA polymers," in *DNA computing and molecular programming*, Springer, 2011, pp. 123–140.
[15] L. F. Landweber and L. Kari, "Universal molecular computation in ciliates," in *Evolution as Computation*, Springer, 2002, pp. 257–274.
[16] L. Kari and L. F. Landweber, "Computational power of gene rearrangement," in *Proceedings of DNA Bases Computers, V American Mathematical Society*, 1999, pp. 207–216.
[17] H. G. Rice, "Classes of recursively enumerable sets and their decision problems," *Trans. Am. Math. Soc.*, pp. 358–366, 1953.
[18] S. Mureev, O. Kovtun, U. T. Nguyen, and K. Alexandrov, "Species-independent translational leaders facilitate cell-free expression," *Nat. Biotechnol.*, vol. 27, no. 8, pp. 747–752, 2009.
[19] G. Zhang, V. Gurtu, and S. R. Kain, "An enhanced green fluorescent protein allows sensitive detection of gene transfer in mammalian cells," *Biochem. Biophys. Res. Commun.*, vol. 227, no. 3, pp. 707–711, 1996.
[20] J. A. Pleiss, M. L. Derrick, and O. C. UHLENBECK, "T7 RNA polymerase produces 5′ end heterogeneity during in vitro transcription from certain templates," *Rna*, vol. 4, no. 10, pp. 1313–1317, 1998.
[21] J. Guhaniyogi and G. Brewer, "Regulation of mRNA stability in mammalian cells," *Gene*, vol. 265, no. 1, pp. 11–23, 2001.
[22] J. Sambrook and D. W. Russell, "Isolation of DNA fragments from polyacrylamide gels by the crush and soak method," *Cold Spring Harb Protoc*, 2006.