Programmable Polymers July 19, 1999

Hilmar Rauhe*, Gaby Voppert, Wolfgang Banzhaf*, Jonathan C. Howardt

DNA molecules can be constructed which allow programmable information processing in vitro. The application of algorithmic rules is achieved by self-assembly of molecules and can be used for DNA computing as well as for the construction of DNA-based nanostructures. The system shown here allows at least for the application of regular grammars to large pools of autonomously computing molecules. It outputs binary data structures which can be read by digital DNA typing and processed further at a single molecule level. As a first application a fast physical random number generator was achieved. Beyond computing, the DNA data structures generated were tested as digital "genetic" fingerprints of various materials such as paint, oil and banknotes.

Since the invention of DNA-based computing[1] most of the algorithms implemented in vitro[1, 2, 3] have tackled NP-complete problems, a class of problems in computer science which is for most problem sizes too complex to be computed exactly with today's algorithms and computers. An alternative to these "hard-wired" approaches is the idea of using self-assembly of DNA molecules as a programmable approach to DNA computing[4]. The aim of programmability is the implementation of various as well as efficient algorithms. Self-assembly led to sophisticated construction of spatial structures of DNA[5] but has not yet been used for computer-like information processing. Though also based on self-assembly, our approach follows a different objective. Instead of the construction of large crystals[5] we focus on parallel computation by large populations of small, autonomous units that can be implemented, read out, isolated and amplified using only common laboratory techniques. For simplicity the approach is restricted to regular grammars, though it can serve as a basis for more sophisticated approaches.

Regular grammars are the simplest formal grammars of the Chomsky hierarchy[6, 7]. Formally a regular grammar is a quadruple G = (Σ, V, R, S) with a finite set Σ of terminals, a finite set V of variables, a start-symbol S and a finite set R of rules r of the form

r: V -> Σ×V ∪ Σ

meaning that a variable of V can be substituted either by a terminal of Σ or by a combination of a terminal and a variable. Here we used the simple grammar G_{rand} = (Σ, V, R, S) with Σ:= {0, 1, s, e}, V:= {A}, S:= S and R:={S:=sA, A->0A, A->1A, A->e}, where 0, 1 represent bits and s, e terminators (start, end).

G_{rand} describes the production of binary strings beginning with s and ending with e and an arbitrary number of bits in between (see Figure 2). Reading the generated strings as binary representations of numbers the grammar describes a random number generator utilizing the principle that the application of A->0A or A->1A in vitro is random.

The rules of the grammar G_{rand} were implemented by "rule-molecules"[4] referred to as "algomers", short double stranded DNA molecules with sticky ends. Each algomer corresponded to exactly one rule of the grammar. Algomers consisted of two annealed complementary oligonucleotides and were constructed such that their double stranded core sequences represented terminals and their sticky ends represented variables (see Figure 1). In the case of bits the oligonucleotides were constructed as "elongators", molecules that can anneal in either direction with defined orientation, and were phosphorylated for subsequent ligation. Assembly of each algomer was done by mixing the corresponding oligonucleotides and annealing them under controlled conditions (see Methods).

Once assembled, the algomers were concatenated by ligation of their sticky ends, thus applying the rules of G_{rand}. This was done in a ligation reaction that was set up as a polymerisation reaction ("bit polymerisation"), which is the true computational step. Polymers yielded from this reaction ("logomers") represent words of the language $L(G_{rand})$, thus are the

* Dept. of Computer Science, University of Dortmund, 44221 Dortmund, Germany
† Institute for Genetics, University of Cologne, 50674 Cologne, Germany

results of the algorithm defined by G_{rand} (see Figure 2 and Figure 3). Under the conditions of our system the populations of logomers after bit polymerisation had about $3x10^{13}$ molecules, their sizes ranging up to 32 bits (see Figure 3). In order to process the result of the computation further the logomers were isolated by ligating them into plasmids and cloning them into bacteria (see Figure 2). In terms of computer science plasmids were used as memory cells and data was written to them by cloning of logomers. The method makes it possible to perform parallel computation at the level of single molecules and to amplify the results of these operations to the macroscopic range by simply growing bacteria.

After isolation, cloned logomers were read out by a digital DNA typing procedure originally developed for minisatellite analysis[8]. The method allowed readout of single logomers directly by PCR. The outputs were binary "printouts" of logomers that could be read by gel-electrophoresis (see Figure 4 and Methods).

Using these methods G_{rand} was used for generation of physical random numbers (see Figure 3 and Figure 4). The performance of G_{rand} was close to that of pseudo random number generators on common hardware (see Figure 6).

Generalizing the method shown for G_{rand} a compiler was developed to translate regular grammars directly to DNA sequences[9]. It allows for the implementation of arbitrary regular grammars. More complex data structures can be implemented using the grammar $G_{dat} = (\Sigma, V, R, S)$, with $\Sigma := \{0, 1, e, s_0, s_1, s_2, \ldots, s_{n-1}, s_n\}$, $V := \{ S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8\}$, start-symbol S and $R := \{S:=s_iS_0, S_0->0S_1, S_0->1S_1, S_1->0S_2, S_1->1S_2, S_2->0S_3, S_2->1S_3, S_3->0S_4, S_3->1S_4, S_4->0S_5, S_4->1S_5, S_5->0S_6, S_5->1S_6, S_6->0S_7, S_6->1S_7, S_7->0S_8, S_7->1S_8, S_8->e$ with $i = \{0, ..., n\}\}$. For $n = 0$ G_{dat} describes a grammar for the generation of all 1-Byte representations of alphanumerical characters, for $n = 4$ the generation of 32-Bit numbers, for arbitrary n the generation of strings of length n.
These data structures are of interest in the context of using logomers as digital DNA fingerprints for labelling of materials. We tested several materials for their ability to be labelled by DNA, fluids such as motor-oil and paint and paper-based materials such as printer paper, 3M Post-Its and banknotes. Logomers were recovered from all tested materials and different concentrations of logomers used to check efficiency of recovery. In comparison to water, paint required 10 fold, oil 100 fold higher concentration (data not shown). Labelling of paper-based materials was highly dependent on consistency of the used paper (see Figure 5).

Decisive in this context is the fact that DNA in itself is not toxic to biological organisms or to the environment, and sequences used for encoding can be constructed which do not contain any biologically relevant information. Since logomers can carry information such as product, manufacturer and expiration date they might be applicable for industrial purpose as well as for characterisation and identification of genetically engineered products such as food. Like the bacteria in our experiments which carry DNA numbers, genetically engineered products could be labelled genomically by recombination techniques[10]. In case of industrial applications our experiments suggest that small amounts of the labelled material should be sufficient for readout, for example from small amounts of car paint in case of accident. If the terminators are kept secret digital DNA-Typing can directly be used for cryptography[11] requiring neither sequencing nor subcloning. In this case the sequence of the readout primers corresponding to start and end are treated as secret keys. Unauthorized access is prevented by adding excess dummy sequences, such as random DNA[11] or logomers with different priming sites. In the context of labelling DNA-based cryptographic methods might become of more than theoretical interest. Encrypted logomers might serve for authenticity checks as well as provide a basis for handling questions of data protection that might arise if the methods shown here lead to digital "genetic" fingerprinting. The techniques used here could also be useful for fabrication of large amounts of DNA-based nanocomponents[5, 12, 13, 14]. Our general approach allows the implementation of more sophisticated algorithms as well as automation of procedures for generation and readout of logomers. Sequencing machines can be used as interface between DNA and silicon computing when reading out the bits like nucleotides. This might lead to a concept of DNA-silicon hybrid computation where standard computers read and process the output of DNA-encoded grammars and in turn initiate further compiling and execution. DNA-silicon interactions will surely become highly significant since it is unlikely that DNA computing will develop very far as a free-standing discipline independent of computing strengths based on silicon.

Methods

Algomer assembly. Oligos were synthesized and PAGE-purified (ARK-Scientific, Darmstadt, Germany). All oligos were dissolved in H_2O 100µM and stored at –20°C. Elongators were phosphorylated, upper and lower strand seperately, in 20µl volume by adding 16µl DNA (100µM), 2µl ligation buffer (NEB) and 2µl PNK (Polynucleotide kinase, NEB) and incubating for 1 hour at 37°C. Algomers were assembled by mixing 20µl of upper and lower strand oligos (each 100µM) and annealing them in a thermocycler (PTC-100, MJ Research) 5' at 95°C and from 75°C to 50°C in steps of 1°C per minute.

Bit polymerisation. Terminators and Elongators were mixed with a ratio of 1:8. In 27µl reaction volume 1µl Start-algomer (50µM), 1µl End-algomer (50µM), 10µl 0-algomer (40µM) and 10µl 1-algomer (40µM), 1,5µl rATP (10mM) and 3,5µl T4 DNA Ligase (NEB) were mixed and incubated 2-3h at 22°C or 24h for benchmarking (see Figure 6).

Isolation. The logomers yielded from bit polymerisation were cloned in pBluescriptIIKS+ (Stratagene). In 10µl reaction volume 5µl pBluescriptIIKS+ (5nM), 0,5µl mix of the bit polymerisation reaction, 1µl ligase buffer (NEB), 0,5µl T4 DNA Ligase (400 NEB units, NEB) and 3µl H_2O were mixed and incubated for 12h at 16°C. 5µl of the reaction mixed was used for subsequent transformation of e.coli DH5-α cells.

DNA typing. Cloned logomers were read out by PCR either directly from a bacterial colony, by dilution of an overnight culture or from dilution of a plasmid preparation. Two PCR reactions were set up each containing the 5' start-primer and either the 3'-0-5' primer or the 3'-1-5' primer. The PCRs result in binary complementary ladder patterns of DNA fragments when visualized by gel-electrophoresis (4% Agarose, stained with 0,001% Ethidiumbromide). Each PCR was prepared in 200µl reaction volume by mixing 144µl H_2O, 20µl PCR Buffer 10x (100 mM Tris-HCl, 500 mM KCl, pH 8.3 at 20°C), 20µl $MgCl_2$ (25mM) , 4µl dNTPs (10mM, Pharmacia Biotech), 2µl Taq-Polymerase (5u/µl, Gibco-BRL), 4µl 5' Primer (10µM), 4µl 3' Primer (10µM) and 4µl logomer cloned in pBluescriptIIKS+ (10^6 molecules/µl) as template. PCR was performed in a thermocycler (PTC-100, MJ Research) using the following protocol: 5' 95°C, 30 cycles of 30'' 95°C, 30'' 69.5°C, 30'' 72°C; stop at 4°C. The results were confirmed by sequencing.

Labelling. 100µl of each fluid (H_2O, motor-oil, paint) was labelled with 1µl 25fM of cloned logomer respectively. After 3 days logomers were re-extracted. Re-extraction from motor-oil (HD 10W-40, TS-Union) was done by adding 100µl of H_2O, thorough vortexing, 5' centrifugation at 20000g and collecting the aequous phase. Re-extraction from paint was done by adding 100µl turpentine, followed by thorough fragmentation and vortexing. After 5' centrifugation on 20000g the supernatant was collected. 100µl H_2O was added and vortexed thoroughly. After 5' centrifugation at 20000g the aequous phase was collected. In the case of paper-based materials 1µl of 25fM cloned logomer was dropped on surface of the paper and left for 3 days. For detection 1mm² of the labelled paper was cut out and used as template directly in a subsequent PCR.

Label detection. 1µl of each extraction volume, or 1mm² paper were used as template in a 50µl PCR reaction containing 35µl H2O, 5µl PCR Buffer 10x, 5µl $MgCl_2$ (25mM), 1µl dNTPs (10mM, Pharmacia Biotech), 0,5µl TAQ (5u/µl, Gibco-BRL), 1µl 5' start-primer (10µM), 1µl 3' end-primer (10µM) using the protocol: 5' 95°C, 30 cycles of 30'' 95°C, 30'' 65°C, 30'' 72°C; stop at 4°C.

Acknowledgements

We thank Libby Guethlein, Ulrich Boehm, Peter Dittrich and Elisabeth C. Proske for discussion and comments; Rita Lange and Sam Saghafi for technical assistance and the people in the lab in Cologne for their friendly support. The work was supported in parts by the Stifterverband der Deutschen Wissenschaft/Stiftung Winterling Marktleuthen and Deutsche Forschungsgemeinschaft.

Correspondence and requests for materials should be addressed to H.R. (rauhe@LS11.informatik.uni-dortmund.de)

References

1. Adleman, L. M. Molecular Computation of Solutions to Combinatorial Problems. *Science*, **266**, 1021-1024, (1994)

2. Guarnieri, F., Fliss, M. , Bancroft, C. Making DNA Add. *Science*, **273**, 220-223, (1996)

3. Ouyang, Q., Kaplan, P. D., Liu, S., Libchaber, A. DNA Solution of the Maximal Clique Problem. *Science*, **278**, 446-449, (1997)

4. Winfree, E. Yang, X., Seeman, N. C. Universal Computation via Self-assembly of DNA: Some Theory and Experiments. *Proceedings of the 2nd DIMACS Meeting on DNA Based Computers, Princeton University, June 20-12*, (1996)

5. Winfree, E., Liu, F., Wenzler, L. A. & Seeman, N. C. Design and self-assembly of two-dimensional DNA crystals. *Nature*, **394**, 539-544, (1998)

6. Chomsky, N. On certain formal properties of grammars, *Inf. and Control.* **2:2**, 137-167, (1959)

7. Hopcroft, J. E., Ullman, J. D. Formal Languages And Their Relation To Automata. Addison-Wesley (1969)

8. Jeffreys, A. J. Minisatellite reapeat coding as a digital approach to DNA typing. *Nature*, **354**, 204-209, (1991)

9. Feldkamp, U., Rauhe, H., (Unpublished results) Dept. of Computer Science. University of Dortmund, Germany.

10. Capecchi M.R. Altering the genome by homologous recombination. *Science*, **244**(4910), 1288-1292, (1989)

11. Clelland C. T., Risca, V., Bancroft, C. Hiding messages in DNA microdots. *Nature*, **399**, 533-534, (1999)

12. Chen, J., Seeman, N.C. Synthesis from DNA of a molecule with the connectivity of a cube. *Nature*, **350**, 631-633, (1991)

13. Braun, E., Eichen Y., Sivan, U., Ben-Yoseph, G. DNA-templated assembly and electrode attachment of an conducting silver wire. *Nature*, **391**, 775-778, (1998)

14. Fink, H.-W., Schönenberger, C. Electrical conduction through DNA molecules. *Nature*, **398**, 407-410, (1999)

15. Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P. Numerical Recipes in C, 2nd Edition, 274-328, Cambridge University Press (1992)

16. The Quantum World Corporation, http://www.comscire.com/QNGINFO.html (1995)

Figure legends

Figure 1: Sequences that were used for implementation of the rules of G_{rand}. Every algomer represents one rule. A->0A and A->1A are elongators and can concatenate in either direction. S:=sA (start) and A->e (end) are terminators and can concatenate only right or left-handed. The double stranded core sequences s, e, 0, 1 represent terminals, the sticky end A represents a variable. H and B on terminators are overhangs compatible to restriction sites (H = HindIII, B = BamHI) and were added for subsequent plasmid cloning. All algomers can only ligate in defined manner and defined direction.

Figure 2: a) Algomers representing the rules of G_{rand}. b) Logomers yielded by bit polymerisation. All logomers are of the form s{0|1}e, consisting of a left-handed bracket s, a right-handed bracket e and an arbitrary number of bits in between. The length of the logomers in terms of algomers can be controlled stochastically by the ratio of elongators to terminators or precisely by the grammar itself if rules are set up to generate polymers of defined size (see text). c) Isolation of single logomers by cloning. The binary “printout” of the cloned logomer can be seen in Figure 4.

Figure 3: Result of a bit polymerisation. The bands show a discrete pattern of logomers of different lengths, each band consisting of a plurality of logomers of same length. The logomers sizes range from 0 bits (60bp) in steps of 30bp per bit to approximately 32 bits (1020bp). Lane 1 shows result of a bit polymerisation of 0s, Lane 2 result of a bit polymerisation of 1s, Lane 3 result of a bit polymerisation of 0s and 1s. Lane 4 contains a 50bp molecular weight marker. Lanes 1 to 3 each contain about 10^{12} random logomers out of a total of 3×10^{13} generated during a single reaction.

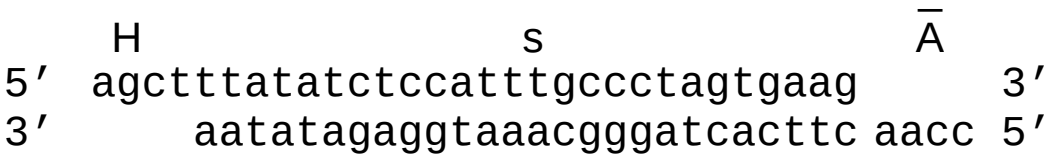
Figure 4: a) Schematic description of digital DNA Typing. Logomer L1 is read out using two PCR reactions. Each reaction contains the start-primer as first and either the 0- or the1-primer as second primer. The resulting complementary pattern can be seen in b). b) Binary "printouts" of three 9-bit random numbers visualized by gel-electrophoresis. Read bottom up, L1 equals $100000110_{bin} = 262_{dec}$, L2 equals $001100001_{bin} = 97_{dec}$, L3 equals $101001001_{bin} = 329_{dec}$. M is a 50bp molecular weight marker.

Figure 5: Labelling of various substances and materials. Lane 1 and 8 are a 50bp weight marker. Lanes 2-7 show PCR products of a 9-bit logomer after recovery from labelled materials. PCR was done with 5’-start and 3’-end primers to detect existence of logomer. Lane 2 shows readout after recovery from H_2O, Lane 3 shows readout after recovery from oil (HD 10W-40, TS-Union), Lane 4 after recovery from paint (Herbol Tafellack, Herbol), Lane 5 after recovery from Post-It (3M), Lane 6 after recovery from printer paper (80g/m², Produkta Etat), Lane 7 after recovery from banknote (10 DM note, Bundesdruckerei, Germany).

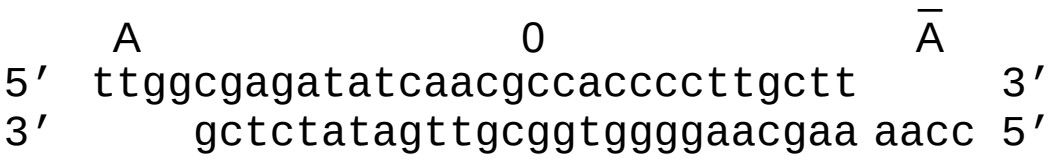
Figure 6: Benchmarking of different random number generators. rng0 – rng4 are pseudo-random number generators as described in Numerical Recipes[15] performed on a PC (AMD K6-II, 450 MHz). QNG is a commercial physical random number generator[16], its speed taken from its specification[16]. G_{rand} was performed as a 24h bit polymerisation using 1400 NEB units T4 DNA Ligase (see Methods). Under these conditions G_{rand} has a theoretical maximum performance of 10^{13} bits per second based on the turnover of T4 DNA-Ligase, thus can be optimized further. Runtime-complexity of the physical random number generators is less than those of the silicon-based pseudo random number generators. Note that G_{rand} is currently missing an automatic I/O system.

Figures and Tables

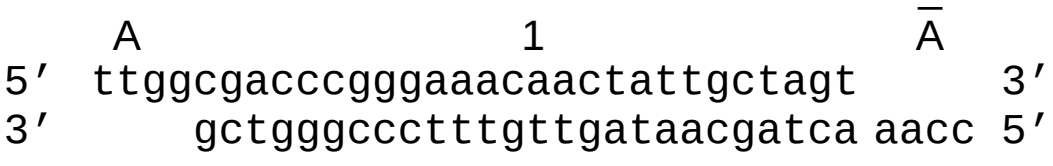
S:=sA



A->0A



A->1A



A->e

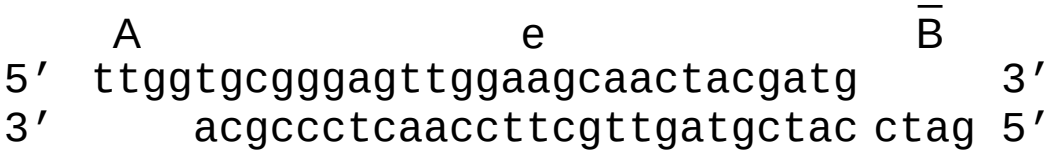


Figure 1

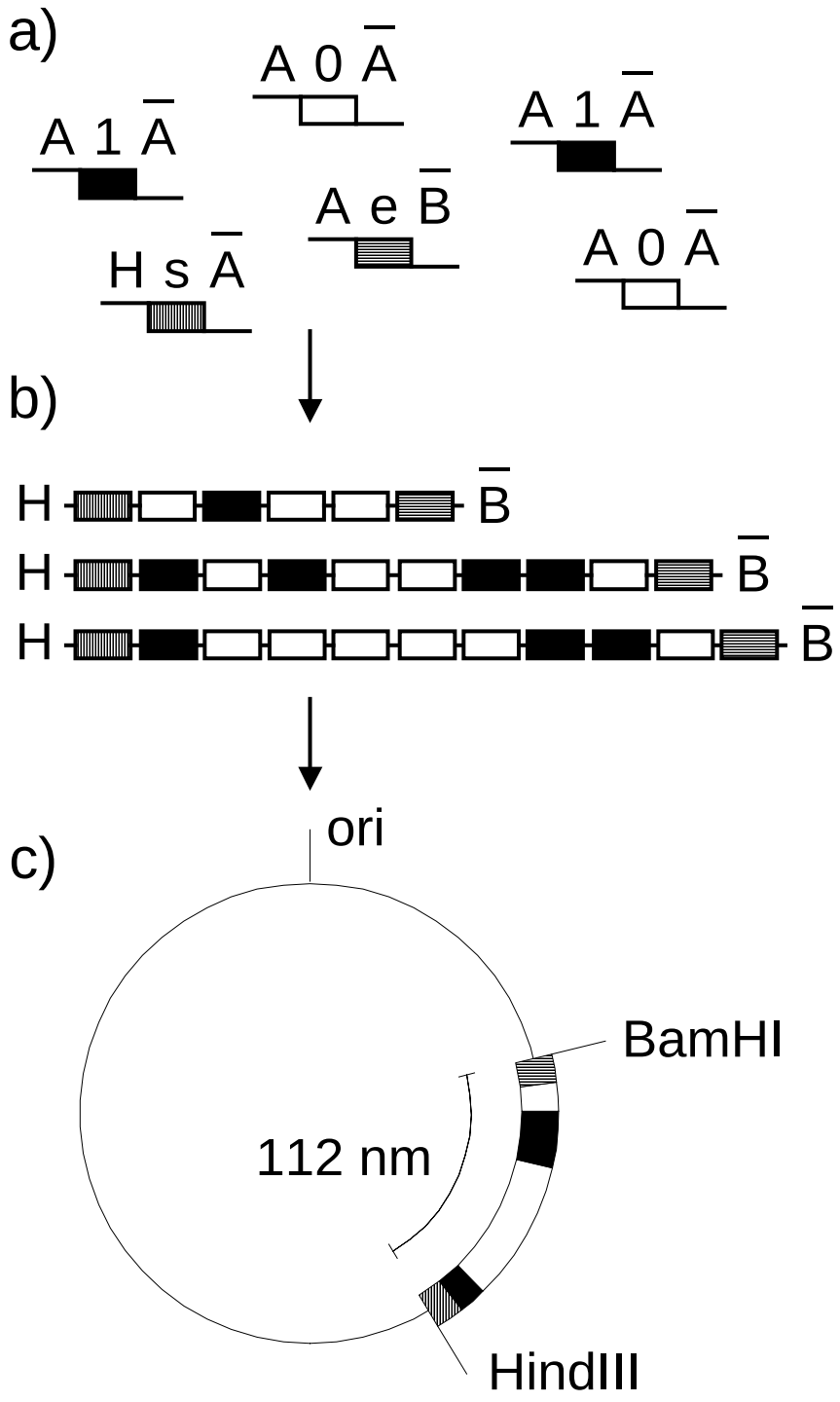


Figure 2

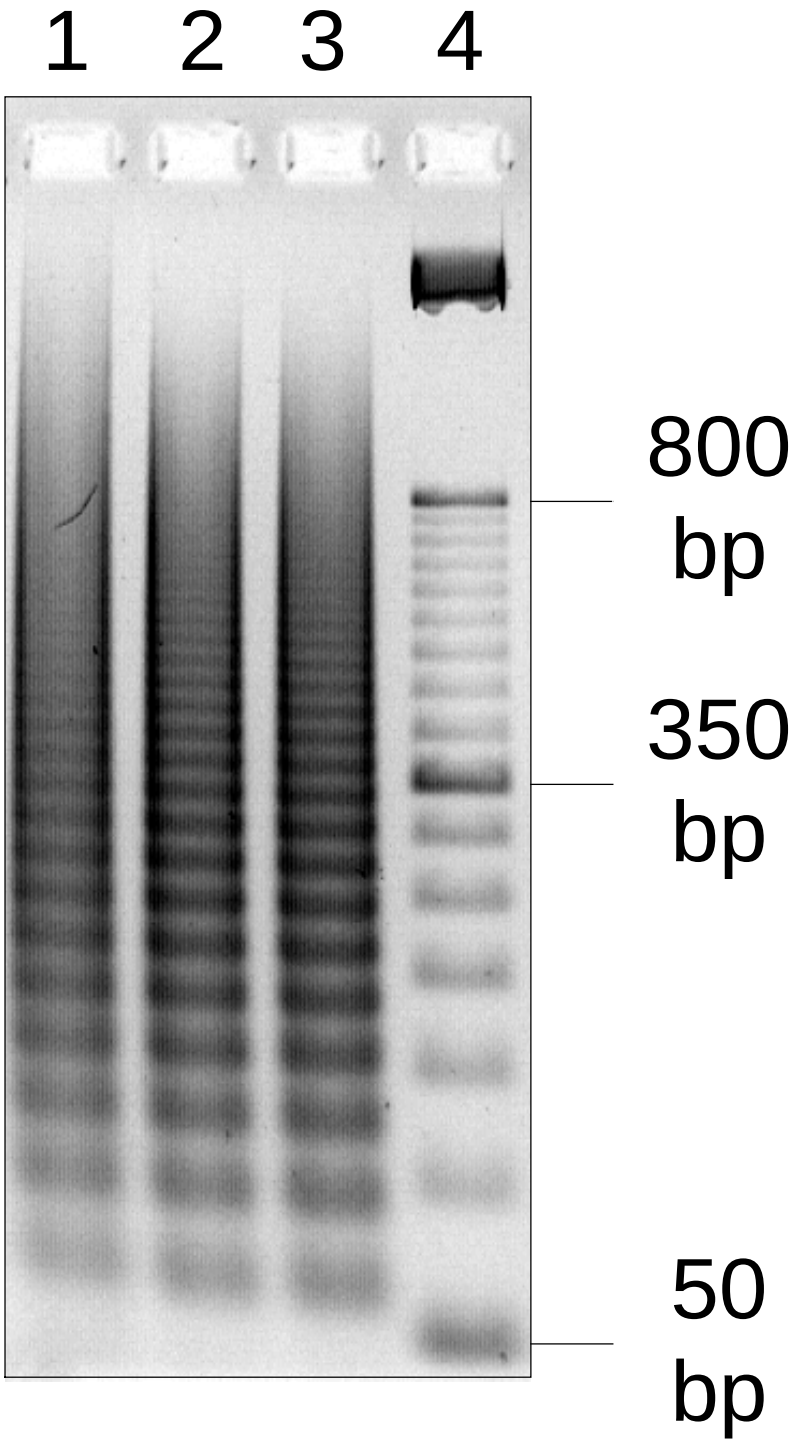


Figure 3

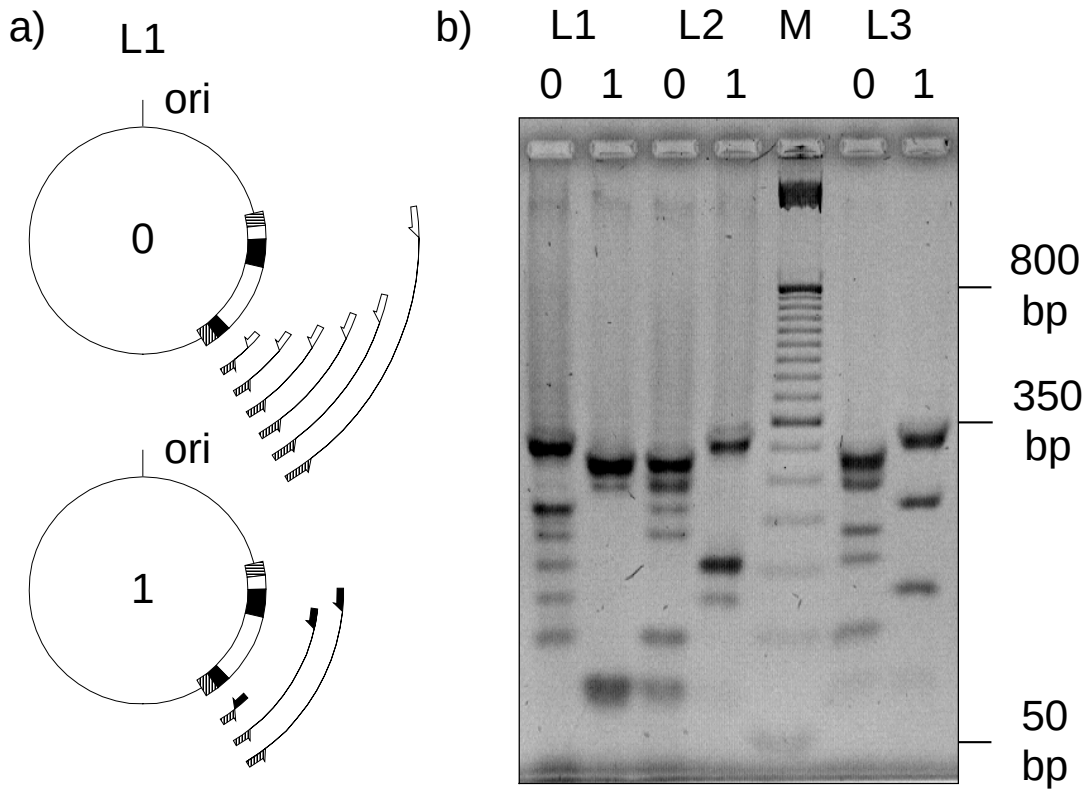


Figure 4

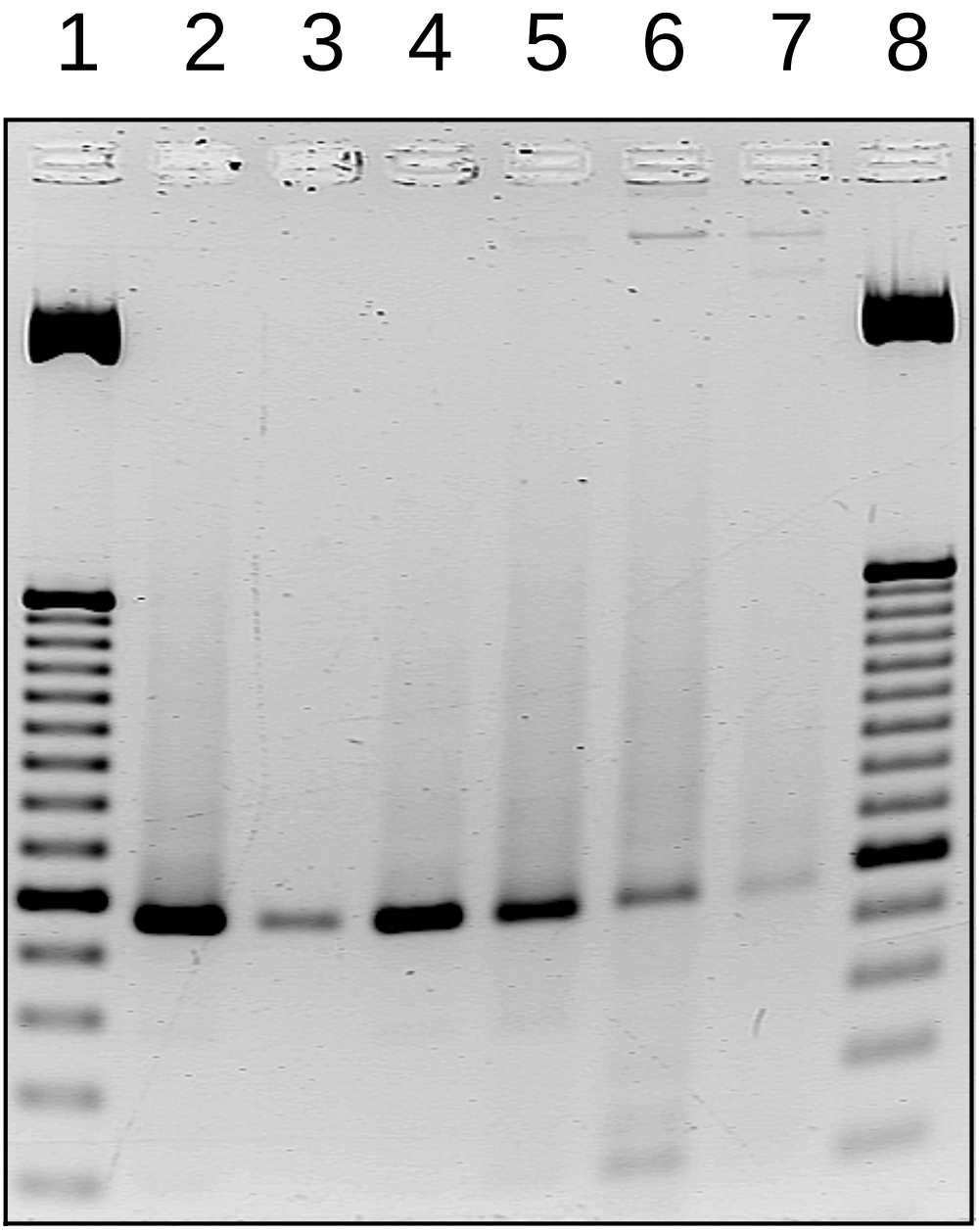


Figure 5

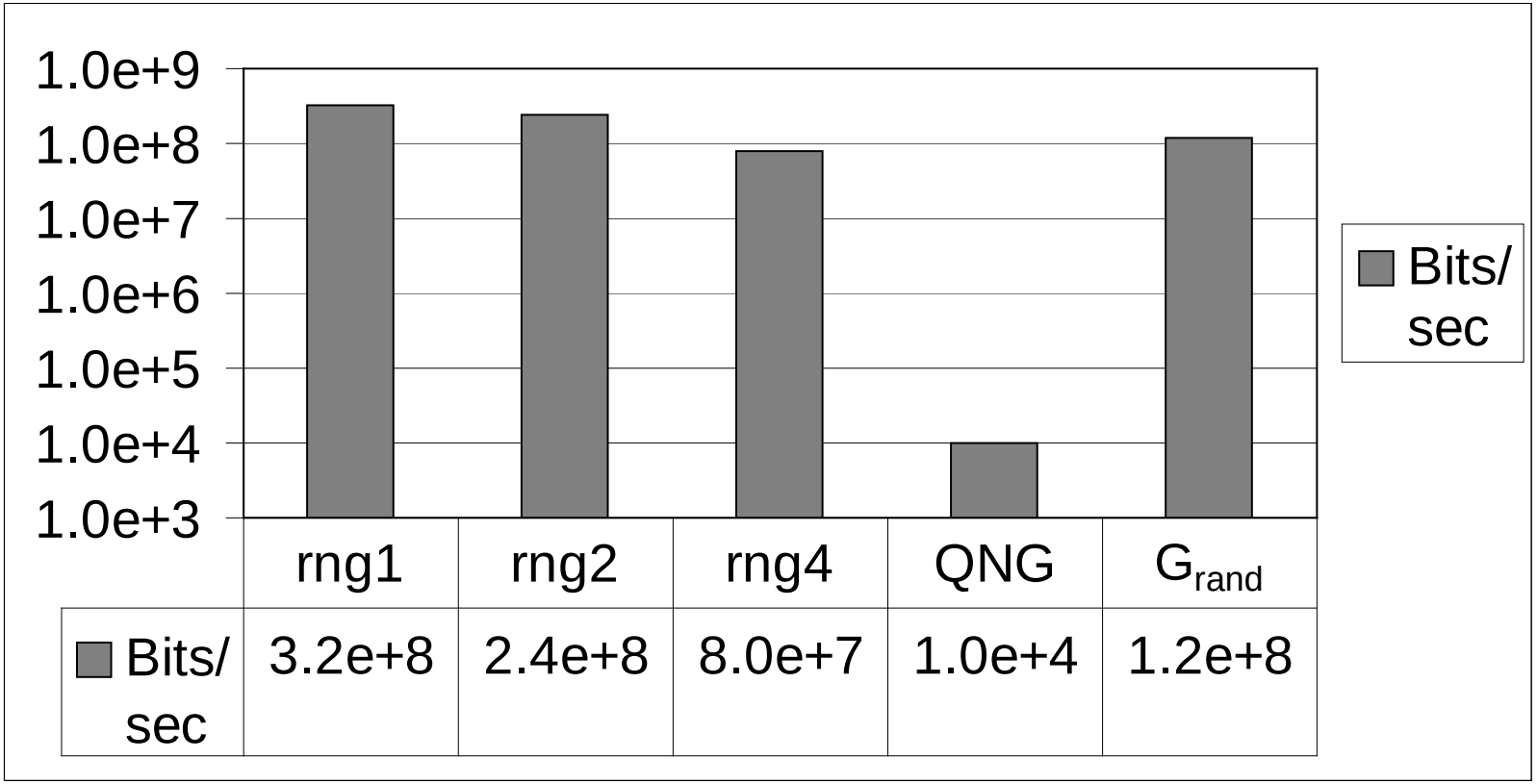

1.0e+9
1.0e+8
1.0e+7
1.0e+6
1.0e+5
1.0e+4
1.0e+3
Bits/ sec
rng1
rng2
rng4
QNG
G rand
Bits/ sec
3.2e+8
2.4e+8
8.0e+7
1.0e+4
1.2e+8

Figure 6