Optimizing nucleic acid sequences for a molecular data recorder

Jerzy Kozyra
Interdisciplinary Computing and Complex Systems Research Group
School of Computing Science, Newcastle University
Newcastle-upon-Tyne, UK NE1 7RU
jurek.kozyra@ncl.ac.uk

Harold Fellermann
Interdisciplinary Computing and Complex Systems Research Group
School of Computing Science, Newcastle University
Newcastle-upon-Tyne, UK NE1 7RU
harold.fellermann@ncl.ac.uk

Ben Shirt-Ediss
Interdisciplinary Computing and Complex Systems Research Group
School of Computing Science, Newcastle University
Newcastle-upon-Tyne, UK NE1 7RU
benjamin.shirt-ediss@ncl.ac.uk

Annunziata Lopiccolo
Interdisciplinary Computing and Complex Systems Research Group
School of Computing Science, Newcastle University
Newcastle-upon-Tyne, UK NE1 7RU
annunziata.lopiccolo@ncl.ac.uk

Natalio Krasnogor
Interdisciplinary Computing and Complex Systems Research Group
School of Computing Science, Newcastle University
Newcastle-upon-Tyne, UK NE1 7RU
natalio.krasnogor@ncl.ac.uk

ABSTRACT

We recently reported the design for a DNA nano-device that can record and store molecular signals. Here we present an evolutionary algorithm tailored to optimising nucleic acid sequences that predictively fold into our desired target structures. In our approach, a DNA device is first specified abstractly: the topology of the individual strands and their desired foldings into multi-strand complexes are described at the domain-level. Initially, this design is decomposed into a set of pairwise strand interactions. Then, we optimize candidate domains, such that the resulting sequences fold with high accuracy into desired target structures both (a) individually and (b) jointly, but also (c) to show high affinity for binding desired partners and simultaneously low affinity to bind with any undesired partner. As optimization heuristic we use a genetic algorithm that employs a linear combination of the above scores. Our algorithm was able to generate DNA sequences that satisfy all given criteria. Even though we cannot establish the theoretically achievable optima (as this would require exhaustive search), our solutions score 90% of an upper bound that ignores conflicting objectives. We envision that this approach can be generalized towards a broad class of toehold-mediated strand displacement systems.

CCS CONCEPTS

•**Computing methodologies** → **Discrete space search;** *Randomized search;* •**Applied computing** → **Chemistry; Physics;**

KEYWORDS

DNA computing; strand displacement; biological data structures; nucleic acid sequence optimization

GECCO '17, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3071178.3071345

ACM Reference format:
Jerzy Kozyra, Harold Fellermann, Ben Shirt-Ediss, Annunziata Lopiccolo, and Natalio Krasnogor. 2017. Optimizing nucleic acid sequences for a molecular data recorder. In *Proceedings of the Genetic and Evolutionary Computation Conference 2017, Berlin, Germany, July 15–19, 2017 (GECCO '17),* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071345

1 INTRODUCTION

Deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) nanotechnology has developed into a vibrant research area with numerous potential applications in molecular computing, bio-manufacturing, and smart therapeutics [19]. At its core, this research exploits the natural Watson-Crick complementarity of DNA and RNA, which causes nucleic acid strands in solution to hybridize spontaneously with complementary regions of the same strand or other strands.

This programmability of nucleic acids has brought about numerous structural nano-devices based on DNA assembly [1, 3, 8], DNA origami [7, 12, 16, 21], and hybrid assemblies where DNA is linked with other functional molecules [9, 11].

DNA nanotechnology can be dynamically functionalized via *toehold-mediated strand displacement* (see grey box on Figure 1). These systems feature short stretches of unpaired, single-stranded nucleotides (referred to as *toeholds*) and DNA strands with partially identical sequences whose competition for common binding partners can induce dynamical shape changes in the DNA/RNA assemblies. Strand displacement causes potentially irreversible structural changes of the nano-device that has been used for programming dynamical behavior such as mechanical actuation [24] and molecular computation [4, 15, 18].

We have recently reported the design for a dynamic DNA nanostructure that implements a molecular signal recorder [10]. This signal recorder is implemented as a linear chain of partially complementary DNA strands which represent data as well as operations. Throughout its operations, the structure exposes a unique binding site, at which it can be commanded—by addition of appropriate strands to the test tube—to either record a signal or to release the

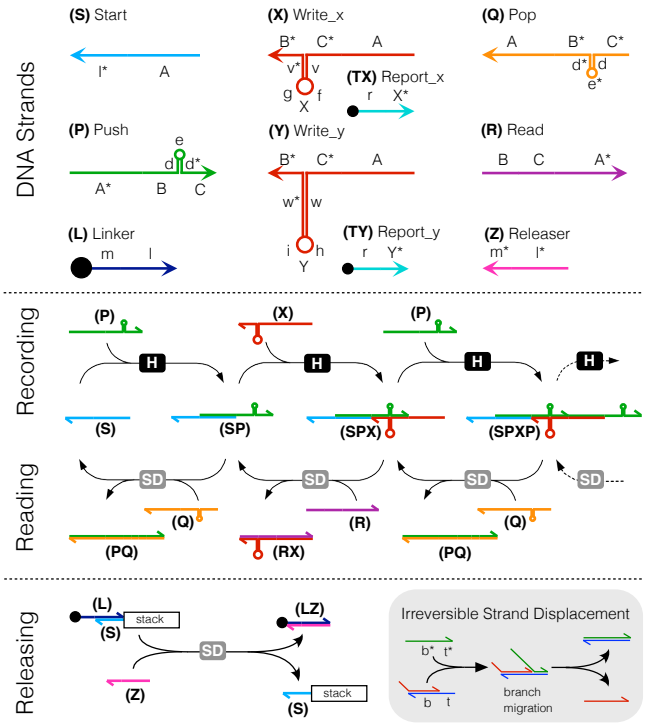

Figure 1: A DNA nano-device that implements a signal recorder via DNA hybridization and strand displacement.

last signal that had been recorded (similar to a stack data structure). In addition to recording and reading, the recorder provides an additional operation to release the entire recorded signal chain from a potential solid support. This operation, as well as two additional reporter strands, have been included for experimental characterization and should not interfere with the principal operation. Figure 1 shows the domain level specification of all ten involved DNA strands of the signal recorder and its three modes of operation. All data and operations are implemented via single stranded DNA strands that interact through DNA hybridization (marked with H in the diagram) and toehold-mediated strand displacement (SD). All processes are designed to be energetically downhill—driven by the binding energy of the closing toehold domains—in order to maximize robustness of the device.

As typical for the discipline, the design in Figure 1 employs an abstract *domain level* specification, where individual nucleic acid strands are resolved down to sequences of not further specified nucleotide stretches (denoted by a, b, c, etc.), rather than at the primary sequence level. In this domain level specification, domains can be specified to be fully complementary (denoted by a^*, b^*, c^*, etc.) and are otherwise assumed to be non-interfering. While the particular nucleotide sequences of the domains are not determined, their lengths are dictated by geometry and energy considerations and are thus part of the domain level specification. Also part of the design are intra-molecular and inter-molecular foldings that the strands should obey once assembled.

Determining nucleic acid sequences that reliably implement a given design is thus an important problem in DNA nanotechnology. Whereas its more prominent inverse problem—to determine the structure a given sequence of nucleobases would fold into—can be solved efficiently and exactly [20], the *nucleic acid design problem* has been proven to be NP-complete [17] and is typically approached using heuristics [5, 26, 27]. Not only is this problem characterized by a vast, discrete, and ragged search space (in our case $4^{208} \approx 1.69 \times 10^{125}$ potential designs), but it is also notoriously difficult to specify what constitutes a "good design".

Most commonly, heuristics employ a set of trial solutions which are scored by some metric that involves secondary structure folding predictions, i.e. solutions to the inverse problem. (The aim of the heuristic is then to iteratively reduce the distance between the given design and the best trial solution). It has been demonstrated that the most successful metrics optimize both *affinity* (a strong tendency of the candidate solution to fold into the target structure) as well as *specificity* (a negligible tendency of the candidate solution to fold into another structure) of the foldings [5, 6].

One noteworthy example of a software suite for the design of nucleic acid structures and devices is NUPACK [22, 23]. At its core, it performs an optimization of the *complex ensemble defect* corresponding to the average number of incorrectly paired nucleotides at equilibrium (evaluated over the ensemble of the test tube). As such, NUPACK ensures the correct folding of the desired complex while minimizing the concentration of undesired "off-target" complexes.

Optimization criteria that are purely based on folding profiles perform very successfully when optimizing nucleotice sequences for individual DNA or RNA foldings, and have even been generalized to operate over pairwise and multistrand foldings [13, 25]. However, by solely considering secondary structure, they fail to address important aspects that emerge when optimizing *systems* of interacting nucleic acid molecules.

In our example signal recorder design, for instance, it is not only important that strands fold into given constructs, but also that hybridization and strand displacement reactions occur with high yield. For if reactions do not proceed to completion, inaccuracies can accumulate over several cycles of signal recording, which might eventually result in the data structure not being able to record any signals, to record too many signals, or to record the wrong signals.

In this article, we propose a novel scoring function for the nucleic acid design problem that applies the criteria of affinity and specificity to the hybridization and branch migration reactions of a DNA nano-device design. After introducing this *favourable equilibrium concentrations* score and demonstrating how it can be incorporated into existing scoring schemes, we perform ensembles of optimization runs with and without this score contribution and present our results, before concluding with a general discussion.

2 SCORING FUNCTIONS FOR NUCLEIC ACID OPTIMIZATION

The signal recorder designs are evaluated based on two factors: desired secondary structure and binding probabilities. We implemented the following partial scoring functions: (i) single-stranded folding S_{sf}, (ii) pairwise folding S_{pf} and (iii) favourable equilibrium concentrations S_{fec}. The first two scores employ secondary structure prediction and a metric that selects for high structure affinity and specificity as discussed in Reference [6]. The third score applies

the same criteria of affinity and specificity to the readiness with which desired or undesired reactions take place.

Single stranded folding. Firstly, we evaluate the ability of a single DNA strand to fold into its specified target structure (as shown in Figure 1 top). Using the secondary structure predictor of the *ViennaRNA* 2.0 software suite [13], we calculate the partition function of all secondary structures that x might fold into.[1]

For any strand x, let $|x|$ denote its length and $\mathbf{d}^x \in \{0,1\}^{|x|}$ a vector whose component $d_i^x = 1$ if base i is specified in the design of x to be bound and 0 otherwise. Further, let $\mathbf{p}^x \in [0,1]^{|x|}$ denote a vector whose i-th component p_i^x denotes the Boltzmann probability (obtained from the partition function) that base i of strand x is paired with another base. The single-stranded folding score S_{sf} is then defined as the normalized Euclidean distance $\|\cdot\|$ between $\mathbf{d}^x$ and $\mathbf{p}^x$ as

$$S_{\text{sf}}(x) = 1 - \frac{1}{|x|}\|\mathbf{d}^x - \mathbf{p}^x\|. \tag{1}$$

Note that $0 \le S_{\text{sf}} \le 1$ and $S_{\text{sf}}(x) = 1$ if x folds unambiguously into its target structure.

Pairwise folding. Secondly, we score the ability of two DNA strands to bind with each other as specified by the design (shown in Figure 1 center and bottom). To obtain this score, we decompose the multi-strand structure of an assembled signal recorder into the set of all its pairwise strand interactions (e.g. SP, PQ, RX).

For each pair x and y of strands, we denote by $\mathbf{d}^{xy} \in \{0,1\}^{|x|+|y|}$ the desired co-folding profile, and calculate—using *ViennaRNA*'s cofold algorithm—the partition function for the pairwise folding of strand x with y, from which we derive the Boltzmann probability vector $\mathbf{p}^{xy} \in [0,1]^{|x|+|y|}$. The pairwise folding score S_{pf} is then defined as

$$S_{\text{pf}}(x,y) = 1 - \frac{1}{|x|+|y|}\|\mathbf{d}^{xy} - \mathbf{p}^{xy}\|. \tag{2}$$

Care has been taken for the push-signal interactions which occur in two variants PX (PY), binding via domains B and C, and XP (YP), binding via domain A. To characterize both variants, the folding predictor has been invoked with constraints that prevent the strands to interact in the respective other domain (e.g. domain A is forbidden to participate in the PX interaction).

Favourable Equilibrium Concentrations. Another requirement for reactions in our signal recorder chemistry is that *desired reactions should be thermodynamically spontaneous* and therefore likely to happen (in the absence of kinetic traps), whereas *undesired cross-reactions should be thermodynamically non-spontaneous* and thus minimized. Maximization of desired reactions thus improves the affinity of the design, whereas minimization of undesired reactions improves its specificity.

A first, simple approach to this scoring function might be:

$$S = -\sum_{i \in \mathcal{R}_{\text{desired}}} w_i \Delta G_i^\circ + \sum_{j \in \mathcal{R}_{\text{undesired}}} w_j \Delta G_j^\circ, \tag{3}$$

which is maximized when (i) the desired reactions each have a maximally negative standard Gibb's free energy change ΔG° and (ii) the undesired reactions each have a close to zero or positive ΔG°. The ΔG° for bimolecular reactions can be calculated by thermodynamic structure prediction algorithms such as *ViennaRNA* or *NUPACK* [25].

The expression for S above is not easy to normalize, however. Moreover, this scoring function also neglects the concentrations of the reaction species involved. At equilibrium in a finite-sized system, the amount that a reaction $A + B \rightleftharpoons AB$ is shifted toward the "left" (to reactants) or toward the "right" (to products) depends on the total concentration of the strands A and B in the system, in addition to the standard Gibb's free energy change ΔG° of the reaction. Taking into account these concerns, developed below is an improved expression for this score.

For a single bimolecular reaction $A + B \rightleftharpoons AB$ in a fixed volume, it can be shown that the equilibrium concentration of product AB, denoted $[AB]_{\text{eq}}$, is the minimum positive solution to the quadratic equation

$$[AB]_{\text{eq}}^2 - \left(C_A + C_B + \frac{1}{K_{\text{eq}}}\right)[AB]_{\text{eq}} + C_A C_B = 0, \tag{4}$$

where $K_{\text{eq}} = \mathrm{e}^{\frac{-\Delta G^\circ}{RT}}$ is the Van't Hoff expression of the reaction equilibrium constant, $C_A = [A]_0 + [AB]_0$ is the conserved total concentration of A strands in the system and $C_B = [B]_0 + [AB]_0$ is the conserved total concentration of B strands. Initial species concentrations are denoted with zero subscripts. Equilibrium concentrations of the A and B strands are respectively:

$$[A]_{\text{eq}} = C_A - [AB]_{\text{eq}} \tag{5}$$

$$[B]_{\text{eq}} = C_B - [AB]_{\text{eq}} \tag{6}$$

Equations (4)-(6) permit to calculate, for a single bimolecular reaction, the equilibrium concentrations of reactant and product species taking into account both ΔG° for the reaction and the total concentration of strands initially present. For this reaction, a 'reaction completion percent' function ε can be defined:

$$\varepsilon(\Delta G^\circ, C_A, C_B) = \frac{2[AB]_{\text{eq}}}{C_A + C_B}, \tag{7}$$

such that $\varepsilon = 1$ when all A,B strands in the system are in the product complex AB at equilibrium, and $\varepsilon = 0$ when all A,B strands in the system are reactants and no product complex is formed. A reaction completion curve may be drawn (Figure 2) denoting reaction ΔG° (x-axis) versus reaction completion percent ε (y-axis), for different values of total strand concentration $C_A + C_B$. Observe that the reaction completion curves of Figure 2 shift to the left as $C_A + C_B$ decreases, and to the right as $C_A + C_B$ increases.

Our signal recorder chemistry, however, does not consist of a single reaction: rather, it consists of a set of interconnected reactions. An analytical expression for the equilibrium point of the whole system is not possible to derive, and so it would appear that function ε cannot be calculated for our signal recorder chemistry. A solution to this dilemma is found by realising that any (well-stirred) system of interconnected bimolecular reactions can be *logically* viewed as a series of separate but communicating single bimolecular reaction sub-systems. As the reaction proceeds, these single bimolecular reaction sub-systems exchange strands, causing the total strand number $C_A + C_B$ in each sub-system to fluctuate. At equilibrium, these fluctuations die out and the global system stabilizes.

[1] Single and pairwise partition functions have been calculated with *ViennaRNA*'s RNAfold and RNAcofold programs, using DNA interaction parameters from Reference [14] at 21°C.

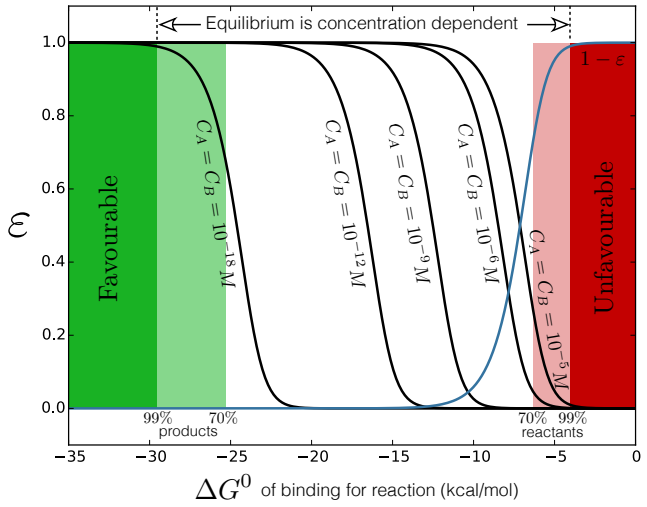

Figure 2: How total strand concentration $C_A + C_B$ affects amount of product at equilibrium for reaction $A + B \rightleftharpoons AB$ at 21°C, when the reaction has different values of Gibb's free energy of binding. Inside the white region, the reaction equilibrium point is concentration dependent. This region is avoided.

Therefore, at equilibrium, we can treat the system as a series of *independent* reactions, and still apply the insights that were developed above (shown in Figure 2) for single bimolecular reactions. The exact equilibrium point of the system need not be calculated. All that needs to be ensured are two weaker conditioins: bimolecular reactions which are 'desirable' should have a ΔG° such that even with a handful of strands (low $C_A + C_B$) the reaction will convert toward 100% products at equilibrium. Then, according to Figure 2, even as $C_A + C_B$ increases, such reaction subsystems will remain at 100% product conversion at equilibrium. Conversely, 'undesirable' bimolecular reactions in our signal recorder chemistry should have ΔG° such that even if all strands in the reaction system were participating in this reaction (high $C_A + C_B$), the reaction will remain as 100% reactants at equilibrium. Then, according to Figure 2, even as $C_A + C_B$ decreases, such reaction subsystems will remain close to 0% product conversion at equilibrium.

Therefore, we define the favourable equilibrium score S_{fec} of a single reaction $r \in \mathcal{R}_{\text{desired}} \cup \mathcal{R}_{\text{undesired}}$ as

$$S_{\text{fec}}(r) = \begin{cases} \varepsilon(\Delta G^\circ_r, C_A^{\text{low}}, C_B^{\text{low}}) & \text{if } r \in \mathcal{R}_{\text{desired}} \\ 1 - \varepsilon(\Delta G^\circ_r, C_A^{\text{high}}, C_B^{\text{high}}) & \text{otherwise.} \end{cases} \quad (8)$$

For our signal recorder chemistry, we set $C_A^{\text{low}} = C_B^{\text{low}} = 10^{-18}$M and $C_A^{\text{high}} = C_B^{\text{high}} = 10^{-5}$M. Note that $0 \le S_{\text{fec}} \le 1$, and that the score approaches 1 when leaving the concentration-dependent zone in Figure 2.

Overall Score Function. To summarize, the three scoring functions were developed to evaluate various aspects of the signal recorder design. Each individual function is normalized to give a score in range $0 \le s \le 1$. However, the function S_{sf} is used to evaluate 13 individual strand structures within the signal recorder design. These strands engage in 25 desired pairwise interactions which are evaluated by function S_{pf}. Finally, function S_{fec} evaluates 86 combinations of pairwise interactions (where 25 are desired and 61 are undesired). This imbalance raises an issue, namely, how does one combine these objective functions? Using a linear combination improves the readability of the results and allows for easy and descriptive comparison between different design aspects.

We established, through several trial runs of the genetic algorithm (described in next section) and upon visual inspection of the obtained results, that the most promising overall score function is given by a linear combination of the scoring functions where each score is weighted by the inverse of the number of factors in its class, *i.e.*, each scoring class is given equal weight. Thus, we define the overall score function as

$$S_{\text{total}} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} S_{\text{sf}}(x) + \frac{1}{|\mathcal{P}|} \sum_{(x,y) \in \mathcal{P}} S_{\text{pf}}(x,y) + \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} S_{\text{fec}}(r), \quad (9)$$

where $\mathcal{S}$ is the set of all specified DNA strands, $\mathcal{P}$ the set of all pairwise specifications, $\mathcal{R} = \mathcal{R}_{\text{desired}} \cup \mathcal{R}_{\text{undesired}}$ the set of all specified reactions, and vertical bars denote set cardinality. We remark that $0 \le S_{\text{total}} \le 3$.

The fine-tuning of the parameters for the overall score function as well as the addition of evaluators for reaction kinetics are left for future research.

3 GENETIC ALGORITHM FOR SEQUENCE OPTIMIZATION

Genetic algorithms (GAs) are a class of heuristics for solving optimization and search problems by mimicking the processes of natural selection. They rely on genetic operators (such as selection, crossover and mutation) to quickly evolve a near-optimal solution for a given objective function. The key advantage of using GAs is that they are effective in navigating a large and complex search space for which little is known.

We based our custom-built GA on the free and open-source *inspyred*[2] framework. The novel elements that we introduced are objective functions (previous section) as well as genetic operators and the design encoding (described below). The source code of our algorithm together with the design specification is available online[3] and the reader is encouraged to examine it.

In our representation, an individual gene encodes the nucleotide sequences of a domain, and the concatenation of all genes forms the genotype of a candidate solution (see Figure 3a). Our code allows to constrain parts of domains to predefined nucleobases (e.g. base sequences recognized by a restriction enzyme), but this feature has not been explored in this study.

The phenotype of a candidate solution is expressed as a complete design of nucleic acid strands assembled from the domains of its genotype (Figure 1). The phenotype is then evaluated using the overall score function S_{total} defined in Equation (9).

The variator combines existing solutions (from the parents population) into other, possibly unexplored solutions that form the

[2] Available at: *https://pypi.python.org/pypi/inspyred*
[3] Available at: *https://bitbucket.org/J3ny/ga-mdr*

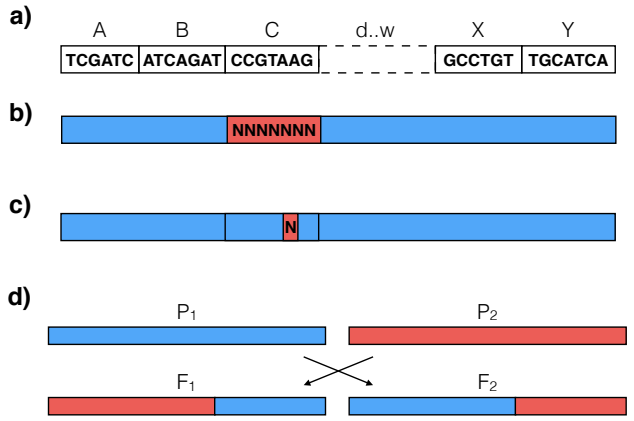

Figure 3: Genotype representation (a) and three genetic operators: (b) domain mutation (c) point mutation (d) crossover.

offspring population. We utilize three genetic operators which are applied to individuals with a certain probability and independently of one another. These are:

- **single-gene mutation:** a gene is picked at random[4] and assigned a random nucleotide sequence – i.e. equivalent to reinitializing the entire domain (with probability 0.02).
- **single-nucleotide mutation:** similar to above, but rather than mutating the entire gene a nucleotide at random position is mutated into another type of nucleotide (with probability 0.25).
- **crossover:** is a standard one-point crossover in which a crossover point is set to a random nucleotide position at the random domain. All nucleotides beyond that point are swapped between the two parents (with probability 0.8).

Through initial experiments with the algorithm parameters (i.e. population size, number of generations) we established 100 individuals over 500 generations to work well. The selector is a default tournament selector; using random sampling it pulls two different individuals from the population and selects one with the higher score. This procedure is repeated until 100 parents are selected for recombination and mutation. In the last step, the replacer discards the worst 2% of the offspring population and retains the top 2% of parents population as survivors (i.e. elite individuals). The evolution is run for 500 generations, and thus the terminator stops the genetic algorithm when a total of 5×10^4 individuals have been evaluated. An individual solution with the highest score is then reported.

The GA was run in two variants: the first variant uses all available partial scoring functions (i.e. S_{sf}, S_{pf} and S_{fec}) for optimization, while the second variant ignores the S_{fec} (the two variants are denoted by (+FEC) and (-FEC) respectively). For comparison, we performed similar optimization of the signal recorder design using *NUPACK*. Each of the three heuristics was run 20 times; yielding 20 different "winning" designs. For additional comparison, we generated 20 solutions where a random sequence is assigned to each domain in the design (referred to as *Random* heuristic).

[4]In our case "picked at random" implies sampling from a discrete uniform distribution (i.e. each outcome is equally likely to happen).

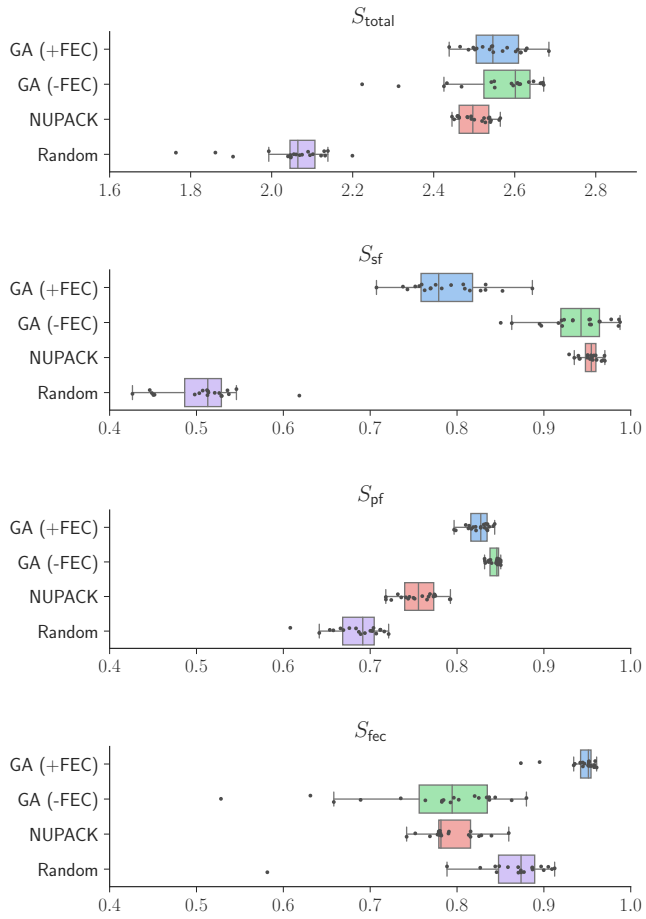

Figure 4: Comparison of the design scores for different heuristics. Each data point is marked with a dot while the boxes show the quartiles of each distribution (whiskers mark the rest of the distribution as a fraction of interquartile range).

4 RESULTS

The results of the 20 highest scoring solutions for each heuristic are shown in Figure 4 (top), together with a decomposition of the total scores into the individual score contributions (below).

At first glance, the three heuristics seem to produce designs of similar quality: the overall score function S_{total} ranges from approximately 2.4 to 2.7 (where the maximum is 3.0) for two GA variants, while *NUPACK* designs are scored slightly lower. For the *Random* heuristic the bulk of the distribution lies above 2.0 which is an interesting result; it implies that a significant part of the score S_{total} may be attributed to the way the design is specified (i.e. with some domains being complementary by construction).

A closer inspection reveals that the heuristics optimize for different objectives. For instance, *NUPACK* excels in S_{sf} and systematically yields high-quality single-stranded foldings, which are

occasionally outmatched only by GA (-FEC). On the other hand, GA (+FEC) does not perform so well in this criterion and has relatively broad S_{sf} distribution, while *Random* is far from optimum. Although the latter is expected, as the random assignment may result in a high ratio of undesired base pairing among random domains, the GA (+FEC) performance is somehow intriguing when compared with GA (-FEC). It indicates that including the function S_{fec} as part of the optimization has a dramatic effect on the individual foldings of strands.

For the scoring function S_{pf} both variants of GA outperform *NUPACK* and *Random*; however, none of the candidate designs considered here has a near-perfect S_{pf} score (unlike for the other scoring functions). Moreover, the *Random* heuristic scores even higher for pairwise folding than for single-stranded folding, which is yet another sign of correct folding "by construction".

Lastly, the scoring function S_{fec}, which evaluates spontaneity of the desired and undesired reactions, is being optimized only when it is explicitly included as part of the optimization heuristic. In comparison, both GA (-FEC) and *NUPACK* have a slightly lower S_{fec} score than the *Random* approach. The relatively high score S_{fec} for *Random* could be explained by the poor folding of individual strands (recall S_{sf} scoring) which potentially leaves fewer bases available for undesired interactions.

We point out that obtaining high scores in S_{sf} and S_{pf} does not automatically translate to a high S_{fec} score (see Figure 5). Our results suggest that the opposite could be true and, in the case of our design, the quality of single-stranded folding may need to be sacrificed for an optimal S_{fec} score.

In order to show that these differences are statistically significant we performed Mann–Whitney U test on pairs of partial scores. In only two cases, the difference in scores is not statistically significant; i.e. the difference between GA (-FEC) and *NUPACK* for S_{fec} score ($p = 0.617$) and S_{sf} score ($p = 0.172$). For all other cases, one heuristic always yields significantly different results ($p \leq 5 \times 10^{-5}$).

The heatmap in Figure 6 decipts the binding energies of all pairwise interactions among all DNA strands of our signal recorder obtained from GA (+FEC) optimization. Desired interactions are marked with a check mark in the table. All other interactions are undesired. Green colors indicate high negative binding energies (strong binding) whereas red coloring indicates weak binding. White coloring indicates regions where S_{fec} is concentration dependent.

It is apparent that our algorithm is able to maximize all desired binding energies and minimize most of the undesired interactions. Note that few interactions failed to be optimized: Start-Read, for example, has a high binding affinity even though a low affinity would be desired. This is because the two strands share a complementary domain (A), which prevents the algorithm from optimizing against this binding, especially since the same domain is responsible for desired binding in other strand pairings. Several interactions (mainly regarding the report strands) remain in the suboptimal concentration dependent region, which is likely due to the short length of these domains. Some pairwise minimum energy folding structures are shown for illustration, of which A1 and A2 are desired, B is undesired and avoidable, whereas C is undesired and unavoidable.

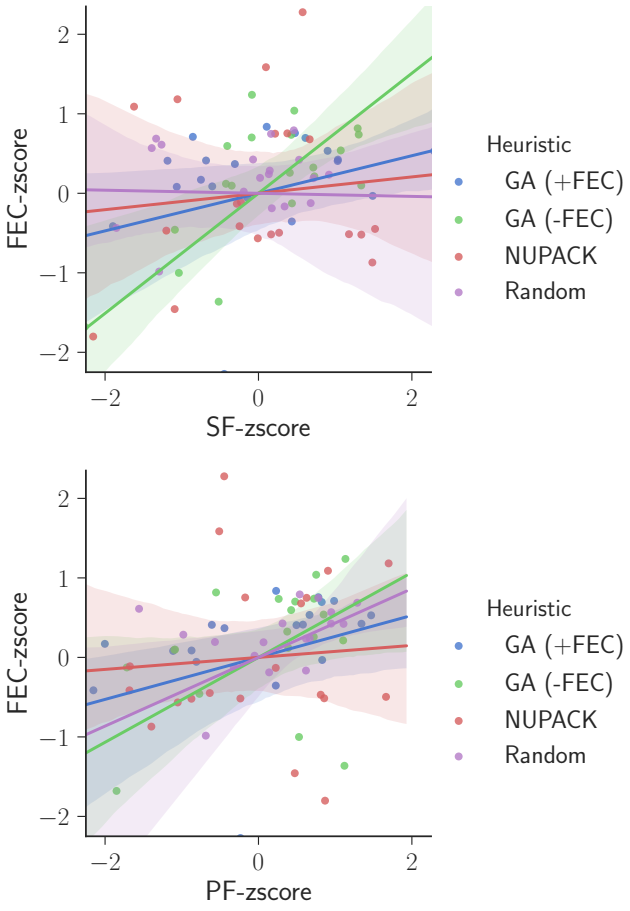

Figure 5: Correlations diagrams with normalised partial scoring functions for different heuristics.

Another aspect, that was not implicitly investigated here, are the manufacturing constraints which are restricting the DNA oligonucleotide synthesis. Even if the *in silico* solution exists, the actual sequence might be extremely difficult to manufacture and purify. In practice, it entails that the final construct has to satisfy the synthesis constraints of a DNA synthesis service. For that instance, guanine-rich sequences are known to form problematic G-quadruplexes [2]. Other considerations typically include identification of homopolymers, interspersed and tandem repeats, and GC-content.

For this reason, we further examined the solutions generated by different heuristics. We discovered that all 20 design produced by GA (-FEC) could not be manufactured - the individual sequences tend to contain patterns of one repeated nucleotide and those repetitions are adjacent to each other (in the worst case 17 consecutive guanine bases). Also, the same problem was encountered during NUPACK optimization which we mitigated by constraining the algorithm to avoid regions of four consecutive nucleotides of the same type.

Interestingly, designs produced by GA (+FEC) did not suffer from the same issue which we assume to be an indirect consequence of S_{fec} optimiztion. In the future, such synthesis constraints should be

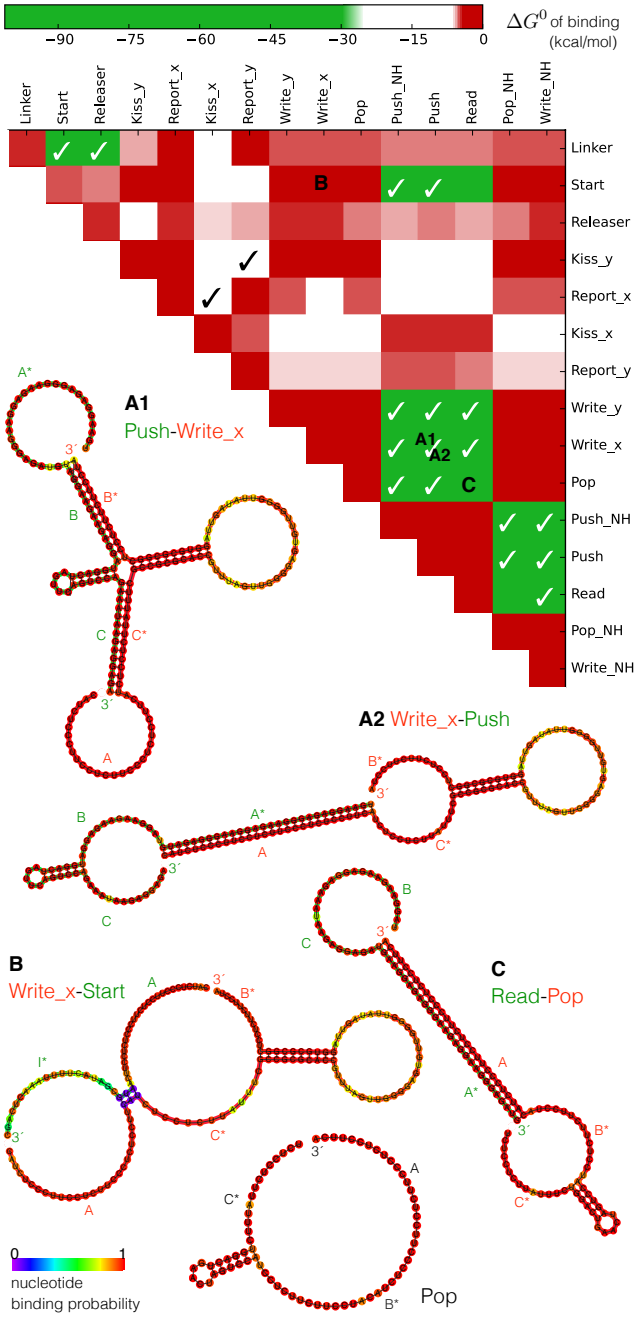

Figure 6: Free energies of binding (top) and example pairwise foldings (bottom) for the sequence-optimized DNA strands, calculated using *ViennaRNA* at 21°C.

incorporated directly into the algorithm as an additional evaluation criterium.

5 DISCUSSION

In this paper, we have used a genetic algorithm to generate nucleic acid sequences that optimize the functioning of a DNA nano-device, namely a molecular signal recorder. Because of the difficulty to determine what constitutes a good design, we have evaluated candidate solutions with multiple score functions, based on individual and pairwise folding, as well as the promiscuity of both desired and undesired reactions. While the approaches based on folding properties of strands are generally acknowledged, methods which guarantee high or low reaction turnover are currently lacking. Yet, this criterion is essential for dynamic nano-devices which require the operation cycle to be strictly and carefully controlled. To the best of our knowledge, our *favourable equilibrium concentration* score is a novel contribution.

We found that the three partial scoring functions are optimizing competing objectives. Ultimately, from the end user point of view, what matters mostly is the best individual, which can then be synthesized and tested in the laboratory. The most promising candidate solution that was produced by the algorithm is evaluated at approximately 90% of the ideal S_{total} score. Although, for this design, the single-stranded folding of individual strands is not optimal, we highlight that for dynamic systems of this kind the self-assembling properties and efficiency of operations of the device are most vital.

We envision that our scoring functions can be used for optimization of nucleic acid sequences for DNA nano-technologies in general, provided that the designs do not involve massive structural rearrangements, in which case the decomposition of the design into pairwise interacting components would not capture important energetic contributions associated with the structural changes.

Future efforts should focus on the inclusion of score functions that evaluate the kinetics of DNA folding and strand displacement to further improve DNA nano-technology designs. Also, one might consider using a multi-objective pareto-based optimization algorithm (or another alternative to the conventional GA) in order to improve the search.

6 ACKNOWLEDGMENTS

This work was supported by grants EP/J004111/2, EP/L001489/2 and EP/N031962/1.

REFERENCES

[1] Dhiraj Bhatia, Shabana Mehtab, Ramya Krishnan, ShantinathfifbS. Indi, Atanu Basu, and Yamuna Krishnan. 2009. Icosahedral DNA Nanocapsules by Modular Assembly. *Angewandte Chemie International Edition* 48, 23 (May 2009), 4134–4137. DOI: http://dx.doi.org/10.1002/anie.200806000

[2] Sarah Burge, Gary N. Parkinson, Pascale Hazel, Alan K. Todd, and Stephen Neidle. 2006. Quadruplex DNA: sequence, topology and structure. *Nucleic Acids Research* 34, 19 (Nov. 2006), 5402–5415. DOI: http://dx.doi.org/10.1093/nar/gkl655

[3] Junghuei Chen and Nadrian C. Seeman. 1991. Synthesis from DNA of a molecule with the connectivity of a cube. *Nature* 350, 6319 (April 1991), 631–633. DOI: http://dx.doi.org/10.1038/350631a0

[4] Y. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, and G. Seelig. 2013. Programmable chemical controllers made from DNA. *Nat. Nano.* 8, 10 (2013), 755–762. DOI: http://dx.doi.org/10.1038/nnano.2013.189

[5] D. C. Dai, H. H. Tsang, and K. C. Wiese. 2009. rnaDesign: Local search for RNA secondary structure design. In *2009 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*. 1–7. DOI: http://dx.doi.org/10.1109/CIBCB.2009.4925700

[6] Robert M. Dirks, Milo Lin, Erik Winfree, and Niles A. Pierce. 2004. Paradigms for computational nucleic acid design. *Nucleic Acids Research* 32, 4 (May 2004), 1392–1403. DOI: http://dx.doi.org/10.1093/nar/gkh291

[7] Shawn M. Douglas, Hendrik Dietz, Tim Liedl, Bjorn Hogberg, Franziska Graf, and William M. Shih. 2009. Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature* 459, 7245 (May 2009), 414–418. DOI: http://dx.doi.org/10.1038/nature08016

[8] Christoph M. Erben, Russell P. Goodman, and Andrew J. Turberfield. 2007. A Self-Assembled DNA Bipyramid. *Journal of the American Chemical Society* 129,

22 (June 2007), 6992–6993. DOI: http://dx.doi.org/10.1021/ja071493b

[9] Harold Fellermann and Luca Cardelli. 2014. Programming chemistry in DNA-addressable bioreactors. *Journal of The Royal Society Interface* 11, 99 (Oct. 2014), 20130987. DOI: http://dx.doi.org/10.1098/rsif.2013.0987

[10] Harold Fellermann, Annunziata Lopiccolo, Jerzy Kozyra, and Natalio Krasnogor. 2016. In Vitro Implementation of a Stack Data Structure Based on DNA Strand Displacement. In *Unconventional Computation and Natural Computation*. Springer, Cham, 87–98. http://link.springer.com/chapter/10.1007/978-3-319-41312-9_8 DOI: 10.1007/978-3-319-41312-9_8.

[11] M. Hadorn, E. Bnzli, H. Fellermann, P. Eggenberger Hotz, and M. Hanczyc. 2012. Specific and reversible DNA-directed self-assembly of emulsion droplets. *Proc. Nat. Acad. Sci. USA* 109, 47 (2012).

[12] Jerzy Kozyra, Alessandro Ceccarelli, Annunziata Lopiccolo, Jing-Ying Gu, Harold Fellermann, Ulrich Stimming, and Natalio Krasnogor. 2017. Designing uniquely addressable bio-orthogonal synthetic scaffolds for DNA and RNA origami. *ACS Synthetic Biology* (2017). submitted.

[13] Ronny Lorenz, Stephan H Bernhart, Christian Hner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. 2011. ViennaRNA Package 2.0. *Algorithms Mol. Biol.* 6, 1 (2011), 26. DOI: http://dx.doi.org/10.1186/1748-7188-6-26

[14] David H. Mathews, Matthew D. Disney, Jessica L. Childs, Susan J. Schroeder, Michael Zuker, and Douglas H. Turner. 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America* 101, 19 (Nov. 2004), 7287–7292. DOI: http://dx.doi.org/10.1073/pnas.0401799101

[15] L. Qian and E. Winfree. 2011. Scaling up digital circuit computation with DNA strand displacement cascades. *Science* 332, 6034 (2011), 1196–201. DOI: http://dx.doi.org/10.1126/science.1200520

[16] Paul W. K. Rothemund. 2006. Folding DNA to create nanoscale shapes and patterns. *Nature* 440, 7082 (March 2006), 297–302. DOI: http://dx.doi.org/10.1038/nature04586

[17] Michael Schnall-Levin, Leonid Chindelevitch, and Bonnie Berger. 2008. Inverting the Viterbi algorithm: an abstract framework for structure design. ACM Press, 904–911. DOI: http://dx.doi.org/10.1145/1390156.1390270

[18] G. Seelig, D. Soloveichik, D. Y. Zhang, and E. Winfree. 2006. Enzyme-Free Nucleic Acid Logic Circuits. *Science* 314, 5805 (2006), 1585–1588. DOI: http://dx.doi.org/10.1126/science.1132493

[19] N. C. Seeman. 2003. DNA in a material world. *Nature* 421, 6921 (2003), 427–431. DOI: http://dx.doi.org/10.1038/nature01406

[20] Michael S Waterman and Temple F Smith. 1986. Rapid dynamic programming algorithms for RNA secondary structure. *Advances in Applied Mathematics* 7, 4 (Dec. 1986), 455–464. DOI: http://dx.doi.org/10.1016/0196-8858(86)90025-4

[21] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. 1998. Design and self-assembly of two-dimensional DNA crystals. *Nature* 394, 6693 (Aug. 1998), 539–544. DOI: http://dx.doi.org/10.1038/28998

[22] Brian R. Wolfe and Niles A. Pierce. 2014. Sequence Design for a Test Tube of Interacting Nucleic Acid Strands. *ACS Synthetic Biology* 4, 10 (oct 2014), 1086–1100. DOI: http://dx.doi.org/10.1021/sb5002196

[23] Brian R. Wolfe, Nicholas J. Porubsky, Joseph N. Zadeh, Robert M. Dirks, and Niles A. Pierce. 2017. Constrained Multistate Sequence Design for Nucleic Acid Reaction Pathway Engineering. *Journal of the American Chemical Society* 139, 8 (feb 2017), 3134–3144. DOI: http://dx.doi.org/10.1021/jacs.6b12693

[24] Bernard Yurke, Andrew J. Turberfield, Allen P. Mills, Friedrich C. Simmel, and Jennifer L. Neumann. 2000. A DNA-fuelled molecular machine made of DNA. *Nature* 406, 6796 (Aug. 2000), 605–608. DOI: http://dx.doi.org/10.1038/35020524

[25] Joseph N. Zadeh, Conrad D. Steenberg, Justin S. Bois, Brian R. Wolfe, Marshall B. Pierce, Asif R. Khan, Robert M. Dirks, and Niles A. Pierce. 2011. NUPACK: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry* 32, 1 (Jan. 2011), 170–173. DOI: http://dx.doi.org/10.1002/jcc.21596

[26] Joseph N. Zadeh, Brian R. Wolfe, and Niles A. Pierce. 2011. Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of Computational Chemistry* 32, 3 (Feb. 2011), 439–452. DOI: http://dx.doi.org/10.1002/jcc.21633

[27] David Yu Zhang. 2010. Towards Domain-Based Sequence Design for DNA Strand Displacement Reactions. In *DNA Computing and Molecular Programming*. Springer, Berlin, Heidelberg, 162–175. http://link.springer.com/chapter/10.1007/978-3-642-18305-8_15 DOI: 10.1007/978-3-642-18305-8_15.