Nucleic Acid Sequence Design via Efficient Ensemble Defect Optimization

JOSEPH N. ZADEH,[1] BRIAN R. WOLFE,[1] NILES A. PIERCE[1,2]
[1]Department of Bioengineering, California Institute of Technology, Pasadena, California, 91125
[2]Department of Applied and Computational Mathematics, California Institute of Technology, Pasadena, California, 91125

Received 4 April 2010; Revised 3 June 2010; Accepted 23 June 2010
DOI 10.1002/jcc.21633
Published online in Wiley Online Library (wileyonlinelibrary.com).

Abstract: We describe an algorithm for designing the sequence of one or more interacting nucleic acid strands intended to adopt a target secondary structure at equilibrium. Sequence design is formulated as an optimization problem with the goal of reducing the ensemble defect below a user-specified stop condition. For a candidate sequence and a given target secondary structure, the ensemble defect is the average number of incorrectly paired nucleotides at equilibrium evaluated over the ensemble of unpseudoknotted secondary structures. To reduce the computational cost of accepting or rejecting mutations to a random initial sequence, candidate mutations are evaluated on the leaf nodes of a tree-decomposition of the target structure. During leaf optimization, defect-weighted mutation sampling is used to select each candidate mutation position with probability proportional to its contribution to the ensemble defect of the leaf. As subsequences are merged moving up the tree, emergent structural defects resulting from crosstalk between sibling sequences are eliminated via reoptimization within the defective subtree starting from new random subsequences. Using a $\Theta(N^3)$ dynamic program to evaluate the ensemble defect of a target structure with N nucleotides, this hierarchical approach implies an asymptotic optimality bound on design time: for sufficiently large N, the cost of sequence design is bounded below by 4/3 the cost of a single evaluation of the ensemble defect for the full sequence. Hence, the design algorithm has time complexity $\Omega(N^3)$. For target structures containing $N \in \{100, 200, 400, 800, 1600, 3200\}$ nucleotides and duplex stems ranging from 1 to 30 base pairs, RNA sequence designs at 37°C typically succeed in satisfying a stop condition with ensemble defect less than $N/100$. Empirically, the sequence design algorithm exhibits asymptotic optimality and the exponent in the time complexity bound is sharp.

© 2010 Wiley Periodicals, Inc. J Comput Chem 000: 000–000, 2010

Key words: RNA; DNA; secondary structure; sequence design; partition function

Introduction

The programmable chemistry of nucleic acid base pairing enables the rational design of self-assembling molecular structures, devices, and systems.[1–3] Here, we address the challenge of encoding equilibrium secondary structure into primary sequence.

Secondary Structure Model

For an RNA strand with N nucleotides, the *sequence*, ϕ, is specified by base identities $\phi_i \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$ for $i = 1, \ldots, N$ (T replaces U for DNA). The *secondary structure*, s, of one or more interacting RNA strands[4] is defined by a set of base pairs (each a Watson Crick pair [A–U or C–G] or wobble pair [G–U]). A *polymer graph* for a secondary structure is constructed by ordering the strands around a circle, drawing the backbones in succession from 5′ to 3′ around the circumference with a nick between each strand, and drawing straight lines connecting paired bases. A secondary structure is *pseudoknotted* if every strand ordering corresponds to a polymer graph with crossing lines. A secondary structure is *connected* if no subset of the strands is free of the others. An *ordered complex* corresponds to the unpseudoknotted structural ensemble, Γ, comprising all connected polymer graphs with no crossing lines for a particular ordering of a set of strands. For a secondary structure, $s \in \Gamma$, the *free energy*,

Additional Supporting Information may be found in the online version of this article.

Correspondence to: N.A. Pierce; e-mail: niles@caltech.edu

Contract/grant sponsor: The National Science Foundation; contract/grant numbers: NSF-CCF-0832824 (The Molecular Programming Project), NSF-CCF-CAREER-0448835
Contract/grant sponsor: The Beckman Institute at Caltech
Contract/grant sponsor: The Ralph M. Parsons Foundation

$\Delta G(\phi, s)$, is calculated using nearest-neighbor empirical parameters for RNA in 1 M Na^+[5,6] or for DNA in user-specified Na^+ and Mg^{++} concentrations.[7–9] This physical model provides the basis for rigorous analysis and design of equilibrium base-pairing in the context of ensemble Γ.

Characterizing Equilibrium Secondary Structure

By calculating the *partition function*[10]

$$Q(\phi) = \sum_{s\in\Gamma} e^{-\Delta G(\phi,s)/k_B T}$$

over Γ, it is possible to evaluate the *equilibrium probability*

$$p(\phi, s) = \frac{1}{Q(\phi)} e^{-\Delta G(\phi,s)/k_B T},$$

of any secondary structure $s \in \Gamma$. Here, k_B is the Boltzmann constant and T is temperature. The secondary structure with the highest probability at equilibrium is the *minimum free energy* (MFE) *structure*,* satisfying

$$s^{\mathrm{MFE}}(\phi) = \arg\min_{s\in\Gamma} \Delta G(\phi, s).$$

The equilibrium structural features of ensemble Γ are quantified by the *base-pairing probability matrix*, $P(\phi)$, with entries $P_{i,j}(\phi) \in [0, 1]$ corresponding to the probability,

$$P_{i,j}(\phi) = \sum_{s\in\Gamma} p(\phi, s) S_{i,j}(s), \qquad (1)$$

that base pair $i \cdot j$ forms at equilibrium. Here, $S(s)$ is a *structure matrix* with entries $S_{i,j}(s) \in \{0, 1\}$. If structure s contains pair $i \cdot j$, then $S_{i,j}(s) = 1$, otherwise $S_{i,j}(s) = 0$. For convenience, the structure and probability matrices are augmented with an extra column to describe unpaired bases. The entry $S_{i,N+1}(s)$ is unity if base i is unpaired in structure s and zero otherwise; the entry $P_{i,N+1}(\phi) \in [0, 1]$ denotes the equilibrium probability that base i is unpaired over ensemble Γ. Hence the row sums of the augmented $S(s)$ and $P(\phi)$ matrices are unity.

The distance between two secondary structures, s_1 and s_2, is the number of nucleotides paired differently in the two structures:

$$d(s_1, s_2) = N - \sum_{\substack{1\le i\le N \\ 1\le j\le N+1}} S_{i,j}(s_1) S_{i,j}(s_2).$$

We also define the discrete delta function

$$\delta_{s_1,s_2} = \begin{cases} 1, & \text{if } d(s_1, s_2) = 0 \\ 0, & \text{otherwise} \end{cases}$$

with respect to secondary structure.

Although the size of the ensemble, Γ, grows exponentially with the number of nucleotides N,[11] the MFE structure, the partition function, and the equilibrium base-pairing probabilities can be evaluated efficiently using $\Theta(N^3)$ dynamic programs.[4, 12–19]

Objective Functions

For a given target structure, s, we formulate sequence design as an optimization problem, minimizing an objective function with respect to sequence, ϕ. Rather than seeking a global optimum, we terminate optimization if the objective function is reduced below a prescribed stop condition.

MFE Defect Optimization

One strategy is to minimize the *MFE defect*:[14, 20–24]

$$\begin{aligned} \mu(\phi, s) &= d(s^{\mathrm{MFE}}, s) \\ &= N - \sum_{\substack{1\le i\le N \\ 1\le j\le N+1}} S_{i,j}(s^{\mathrm{MFE}}(\phi)) S_{i,j}(s), \end{aligned}$$

corresponding to the distance between the MFE structure $s^{\mathrm{MFE}}(\phi)$ and the target structure s. The utility of this approach hinges on whether or not the equilibrium structural features of ensemble Γ are well-characterized by the single structure $s^{\mathrm{MFE}}(\phi)$, which in turn depend on the specific sequence ϕ.[24] If $\mu(\phi, s) = 0$, the target structure s is the most probable secondary structure at equilibrium; $p(\phi, s)$ can nonetheless be arbitrarily small, because of the competition from other secondary structures in Γ.

Probability Defect Optimization

To address this concern, an alternative strategy is to minimize the *probability defect*:[14, 24–26]

$$\pi(\phi, s) = 1 - p(\phi, s),$$

corresponding to the sum of the probabilities of all nontarget structures in the ensemble Γ. If $\pi(\phi, s) \approx 0$, the sequence design is essentially ideal because the equilibrium structural properties of the ensemble are dominated by the target structure s. However, as $\pi(\phi, s)$ deviates from zero, it increasingly fails to characterize the quality of the sequence because the probability defect treats all nontarget structures as being equally defective. This property is a concern for challenging designs, where it may be infeasible to achieve $\pi(\phi, s) \approx 0$.

Ensemble Defect Optimization

To address these shortcomings, a third strategy is to minimize the *ensemble defect*:[24]

$$\begin{aligned} n(\phi, s) &= \sum_{\sigma\in\Gamma} p(\phi, \sigma) d(\sigma, s) \\ &= N - \sum_{\substack{1\le i\le N \\ 1\le j\le N+1}} P_{i,j}(\phi) S_{i,j}(s), \end{aligned}$$

*For simplicity of exposition, we assume that there is a unique MFE structure; only superficial changes are required if this is not the case.

corresponding to the average number of incorrectly paired nucleotides at equilibrium calculated over ensemble Γ.

Comparing Formulations

We cast these three objective functions into a unified formulation to highlight their differences:

$$n(\phi, s) = \sum_{\sigma\in\Gamma} p(\phi, \sigma) d(\sigma, s),$$

$$\mu(\phi, s) = \sum_{\sigma\in\Gamma} \delta_{\sigma,s^{\mathrm{MFE}}} d(\sigma, s),$$

$$\pi(\phi, s) = \sum_{\sigma\in\Gamma} p(\phi, \sigma)(1 - \delta_{\sigma,s}).$$

Using $n(\phi, s)$ to perform ensemble defect optimization, the average number of incorrectly paired nucleotides at equilibrium is evaluated over ensemble Γ using $p(\phi, \sigma)$, the Boltzmann-weighted probability of each secondary structure $\sigma \in \Gamma$, and $d(\sigma, s)$, the distance between each secondary structure $\sigma \in \Gamma$ and the target structure s. By comparison, using $\mu(\phi, s)$ to perform MFE defect optimization, $p(\phi, \sigma)$ is replaced by the discrete delta function $\delta_{\sigma,s^{\mathrm{MFE}}}$, which is unity for s^{MFE} and zero for all other structures $\sigma \in \Gamma$. Alternatively, using $\pi(\phi, s)$ to perform probability defect optimization, $d(\sigma, s)$ is replaced by the binary distance function $(1 - \delta_{\sigma,s})$, which is zero for s and 1 for all other structures $\sigma \in \Gamma$. Hence, the MFE defect makes the optimistic assumption that s^{MFE} will dominate Γ at equilibrium, whereas the probability defect makes the pessimistic assumption that all structures $\sigma \in \Gamma$ with $d(\sigma, s) \neq 0$ are equally distant from the target structure s. The objective function $n(\phi, s)$ quantifies the equilibrium structural defects of sequence ϕ even when $\mu(\phi, s)$ and $\pi(\phi, s)$ do not. In the present work, we perform ensemble defect optimization by minimizing $n(\phi, s)$ with respect to ϕ.

Prior Optimization Algorithms

Previous implementations of probability defect optimization[14,24,25] and ensemble defect optimization[24] employed single-scale mutation procedures in which each candidate mutation was evaluated on the full sequence using a $\Theta(N^3)$ dynamic program to calculate $Q(\phi)$ or $P(\phi)$, respectively. By comparison, more efficient hierarchical mutation procedures have been developed for MFE defect optimization.[14,20–22] These methods perform a hierarchical decomposition of the target structure, optimizing subsequences on a series of growing substructures to reduce the number of times that $s^{\mathrm{MFE}}(\phi)$ is calculated on the full sequence using a $\Theta(N^3)$ dynamic program. Furthermore, to reduce the total number of mutations that must be evaluated, these methods guide the selection of candidate mutation positions based on defects in the MFE substructure.[14,20–22]

Algorithm

Here, we describe a sequence design algorithm that achieves high design quality via ensemble defect optimization, and low design cost via hierarchical structure decomposition and defect-weighted sampling. For a given target secondary structure, s, with N nucleotides, we seek to design a sequence, ϕ, with ensemble defect, $n(\phi, s)$, satisfying the *stop condition*:

$$n(\phi, s) \leq f_{\mathrm{stop}} N,$$

for a user-specified value of $f_{\mathrm{stop}} \in (0, 1)$. Candidate mutations are evaluated at the leaves of a binary tree decomposition of the target structure. During leaf optimization, defect-weighted mutation sampling is used to select each candidate mutation position with probability proportional to its contribution to the ensemble defect of the leaf. If emergent structural defects are encountered when merging subsequences moving up the tree, they are eliminated via defect-weighted child sampling and reoptimization. This design algorithm is outlined below and detailed in the pseudocode of Algorithm 1 (see Appendix).

Hierarchical Structure Decomposition

Prior to sequence design, the target structure s is decomposed into a (possibly unbalanced) binary tree of substructures, with each node of the tree indexed by a unique integer k. For each parent node, k, there is a left child node, k_l, and a right child node, k_r. Each nucleotide in parent structure s^k is partitioned to either the left or right child substructure ($s^k = s^k_l \cup s^k_r$ and $s^k_l \cap s^k_r = \emptyset$). Child node k_l inherits from parent node k the augmented substructure, s^k_{l+}, comprising *native nucleotides*, $s^{k_l}_{\mathrm{native}} \equiv s^k_l$, and additional *dummy nucleotides* that approximate the influence of its sibling in the context of their parent ($s^{k_l} \equiv s^{k_l}_{\mathrm{native}} \cup s^{k_l}_{\mathrm{dummy}} \equiv s^k_{l+}$).

In contrast to earlier hierarchical methods that decompose parent structures at multiloops,[14,20,22] our algorithm decomposes parent structures within duplex stems. This approach is more generally applicable to the design of duplex-rich engineered structures that often contain no multiloops. Eligible split-points are those locations within a duplex stem with at least H_{split} consecutive base-pairs to either side, such that both children would have at least N_{split} nucleotides. If there are no eligible split-points, a structure becomes a leaf node in the decomposition tree. Otherwise, an eligible split-point is selected so as to minimize the difference in the size of the children, $||s^k_l| - |s^k_r||$. Dummy nucleotides are defined by extending the newly-split duplex stem across the split-point by H_{split} base pairs ($|s^{k_l}_{\mathrm{dummy}}| = 2H_{\mathrm{split}}$). See Figure 1 for an example of a hierarchical structure decomposition.

For a parent node k, the sequence ϕ^k follows the same partitioning as the structure s^k ($\phi^k = \phi^k_l \cup \phi^k_r$ and $\phi^k_l \cap \phi^k_r = \emptyset$). Likewise, for a child node k_l, the sequence contains both native and dummy nucleotides ($\phi^{k_l} \equiv \phi^{k_l}_{\mathrm{native}} \cup \phi^{k_l}_{\mathrm{dummy}} \equiv \phi^k_{l+}$).

For any node k with sequence ϕ^k and structure s^k, the ensemble defect, $n^k \equiv n(\phi^k, s^k)$, may be expressed as

$$n^k = \sum_{1\leq i\leq |s^k|} n^k_i,$$

where

$$n^k_i = 1 - \sum_{1\leq j\leq |s^k|+1} P^k_{i,j} S^k_{i,j}$$

is the contribution of nucleotide i to the ensemble defect of the node. For a parent node k, the ensemble defect can be expressed as a sum of contributions from bases partitioned to the left and right children ($n^k = n^k_l + n^k_r$). For a child node k_l, the ensemble defect can be expressed as a sum of contributions from native and dummy nucleotides ($n^{k_l} = n^{k_l}_{\text{native}} + n^{k_l}_{\text{dummy}}$). Conceptually, $n^{k_l}_{\text{native}}$, the contribution of the native nucleotides to the ensemble defect of child k_l [calculated on child node k_l at cost $\Theta(|s^{k_l}|^3)$], approximates n^k_l, the contribution of the left-child nucleotides to the ensemble defect of parent k [calculated on parent node k at higher cost $\Theta(|s^k|^3)$]. In general, $n^{k_l}_{\text{native}} \neq n^k_l$, because the dummy nucleotides in child node k_l only approximate the influence of its sibling (which is fully accounted for only in the more expensive calculation on parent node k).

The utility of hierarchical structure decomposition hinges on the assumption that sequence space is sufficiently rich that two subsequences optimized for sibling substructures will often not exhibit crosstalk when merged by a parent node. Our hierarchical mutation procedure is designed to benefit from this property when it holds true, and to eliminate emergent defects when they do arise.

Leaf Optimization with Defect-Weighted Mutation Sampling

The sequence design process is *initialized* by randomly specifying the identities of all nucleotides in the leaf structures, subject to the constraint that bases intended to be paired are chosen to be Watson-Crick complements. At leaf node k, sequence optimization is performed by mutating either one base at a time (if $S^k_{i,|s^k|+1} = 1$) or one base pair at a time (if $S^k_{i,j} = 1$ for some $1 \leq j \leq |s^k|$, in which case ϕ^k_i and ϕ^k_j are mutated simultaneously so as to remain Watson-Crick complements).

We perform *defect-weighted mutation sampling* by selecting nucleotide i as a candidate for mutation with probability n^k_i/n^k. A candidate sequence $\hat{\phi}^k$ is evaluated via calculation of $\hat{n}^k$ if the candidate mutation, ξ, is not in the set of previously rejected mutations, $\gamma_{\text{unfavorable}}$ (position and sequence). A candidate mutation is retained if $\hat{n}^k < n^k$ and rejected otherwise. The set, $\gamma_{\text{unfavorable}}$, is updated after each unsuccessful mutation and cleared after each successful mutation.

Optimization of leaf k terminates successfully if the *leaf stop condition*:

$$n^k \leq f_{\text{stop}}|s^k|$$

is satisfied, or restarts if $M_{\text{unfavorable}}|s^k|$ consecutive unfavorable candidate mutations are either in $\gamma_{\text{unfavorable}}$ or are evaluated and added to $\gamma_{\text{unfavorable}}$. Leaf optimization is restarted from new random initial conditions up to M_{leafopt} times before terminating unsuccessfully. The outcome of leaf optimization is the leaf sequence ϕ^k corresponding to the lowest encountered value of the leaf ensemble defect n^k.

Subsequence Merging and Reoptimization

After sibling nodes k_l and k_r have been optimized, parent node k merges their native subsequences (setting $\phi^k_l = \phi^{k_l}_{\text{native}}$ and $\phi^k_r = \phi^{k_r}_{\text{native}}$) and evaluates n^k to check the *parental stop condition*:

$$n^k \leq \max\left(f_{\text{stop}}|s^k_l|, n^{k_l}_{\text{native}}\right) + \max\left(f_{\text{stop}}|s^k_r|, n^{k_r}_{\text{native}}\right).$$

If this stop condition is satisfied, subsequence merging continues up the tree. Otherwise, failure to satisfy the stop condition implies the existence of *emergent defects* resulting from crosstalk between the two child sequences. In this case, parent node k initiates *defect-weighted child sampling* and reoptimization within its subtree. Left child k_l is selected for reoptimization with probability n^k_l/n^k and right child k_r is selected for reoptimization with probability n^k_r/n^k. This defect-weighted child sampling procedure is performed recursively until a leaf is encountered (each time using partitioned defect information inherited from the parent k that initiated the reoptimization). The standard leaf optimization procedure is then performed starting from a new random initial sequence. The use of random initial conditions during leaf reoptimization is based on the assumption that sequence space is sufficiently rich that emergent defects can typically be eliminated simply by designing a different leaf sequence. Following leaf reoptimization, merging begins again starting with the reoptimized leaf and its sibling. The elimination of emergent defects in parent k by defect-weighted child sampling and reoptimization is attempted up to M_{reopt} times.

Optimality Bound and Time Complexity

This hierarchical sequence design approach implies an asymptotic optimality bound on the cost of designing the full sequence relative to the cost of evaluating a single candidate mutation on the full sequence. For a target structure with N nucleotides, evaluation of a candidate sequence requires calculation of $n(\phi, s)$ at cost $c_{\text{eval}}(N) = \Theta(N^3)$. Performing sequence design using hierarchical structure decomposition, mutations are evaluated at the leaf nodes and merged subsequences are evaluated at all other nodes. For node k, the evaluation cost is $c_{\text{eval}}(|s^k|)$. If at least one mutation is required in each leaf, the design cost is minimized by maximizing the depth of the binary tree. Furthermore, at each depth in the tree, the design cost is minimized by balancing the tree. Hence, a lower bound on the cost of designing the full sequence is given by

$$c_{\text{des}}(N) \geq c_{\text{eval}}(N)\left[1 + 2\left(\frac{1}{2}\right)^3 + 4\left(\frac{1}{4}\right)^3 + 8\left(\frac{1}{8}\right)^3 + \ldots\right]$$

or

$$c_{\text{des}}(N) \geq \frac{4}{3}c_{\text{eval}}(N).$$

Hence, if the sequence design algorithm performs optimally for large N, we would expect the cost of full sequence design to be 4/3 the cost of evaluating a single mutation on the full sequence. In practice, many factors might be expected to undermine optimality: imperfect balancing of the tree, the addition of dummy nucleotides in each non-root node, the use of finite tree depth, leaf optimizations requiring evaluation of multiple candidate mutations, and reoptimization to eliminate emergent defects. This optimality bound implies time complexity $\Omega(N^3)$ for the sequence design algorithm.

Methods

Computational sequence design studies were performed using the default algorithm parameters of Table 1. Design trials were run on

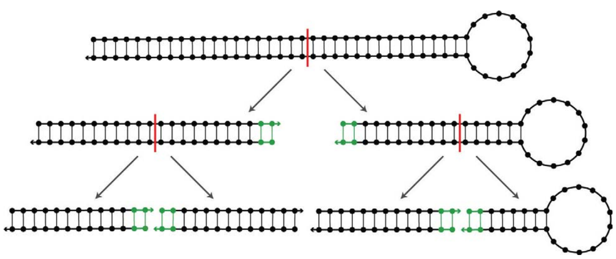

Figure 1. Hierarchical decomposition of a target structure. The split-point within each parent structure is denoted by a red line. The dummy nucleotides within each child structure are depicted in green. The native nucleotides within each structure are depicted in black. $H_{\text{split}} = 2, N_{\text{split}} = 20$.

a cluster of 2.53 GHz Intel E5540 Xeon dual-processor/quad-core nodes with 24 GB of memory per node.

Target Structure Test Sets

Algorithm performance was evaluated on structure test sets containing 30 target structures for each of $N \in \{100, 200, 400, 800, 1600, 3200\}$. An *engineered test set* was generated by randomly selecting structural components and dimensions from ranges intended to reflect current practice in engineering nucleic acid secondary structures. A multi-stranded version was produced by introducing nicks into the structures in the engineered test set. Each structure in a *random test set* was obtained by calculating an MFE structure of a different random RNA sequence at 37°C. Figure 2 compares the structural features of the engineered and random test sets. In general, the random test set has target structures with a lower fraction of bases paired, more duplex stems, and shorter duplex stems (as short as one base pair). Additional structural features of the engineered and random test sets are summarized in Supporting Information Figure S1. The structure test sets are available as Supporting Information. For the design studies that follow, new target structure test sets were generated from scratch. The design algorithm was not tested on these structures prior to generating the depicted results.

Other Algorithms

To illustrate the roles of hierarchical structure decomposition and defect-weighted sampling in the context of ensemble defect optimization, we compare our algorithm to three alternative algorithms lacking either or both of these features:

Table 1. Default Parameter Values Used in Evaluating Algorithm Performance for RNA Design.

Parameter	Value
H_{split}	2
N_{split}	20
f_{stop}	0.01
M_{reopt}	10
M_{leafopt}	3
$M_{\text{unfavorable}}$	4

For DNA design, $H_{\text{split}} = 3$.

- *Single-scale ensemble defect optimization with uniform mutation sampling*:[24] The leaf optimization algorithm is applied directly on the full sequence using *uniform mutation sampling* in which each candidate mutation position is selected with equal probability (pseudocode in Supporting Information Algorithm S1).
- *Single-scale ensemble defect optimization with defect-weighted mutation sampling*: The leaf optimization algorithm is applied directly on the full sequence (pseudocode in Supporting Information Algorithm S2).
- *Hierarchical ensemble defect optimization with uniform sampling*: The hierarchical algorithm is applied using uniform mutation sampling during leaf optimization and uniform child sampling during subsequence merging and reoptimization (pseudocode in Supporting Information Algorithm S3).

We also modified our algorithm to compare performance to algorithms inspired by previous work:

- *Single-scale probability defect optimization with uniform mutation sampling*:[14,24–26] This method seeks to design a sequence such that the probability defect satisfies the stop condition $\pi(\phi, s) \leq f_{\text{stop}}$. For $f_{\text{stop}} \in (0, 0.5]$, satisfaction of this stop condition is sufficient to ensure that stop conditions $n(\phi, s) \leq f_{\text{stop}}N$ and $\mu(\phi, s) \leq f_{\text{stop}}N$ are also satisfied. Optimization is performed using a modified version of the leaf optimization algorithm (with $\pi(\phi, s)$ taking the role of $n(\phi, s)$) applied directly on the full sequence using uniform mutation sampling (pseudocode in Supporting Information Algorithm S4).
- *Hierarchical MFE defect optimization with defect-weighted sampling*:[14,20–22] This method seeks to design a sequence such that the MFE defect satisfies the stop condition $\mu(\phi, s) \leq f_{\text{stop}}N$.

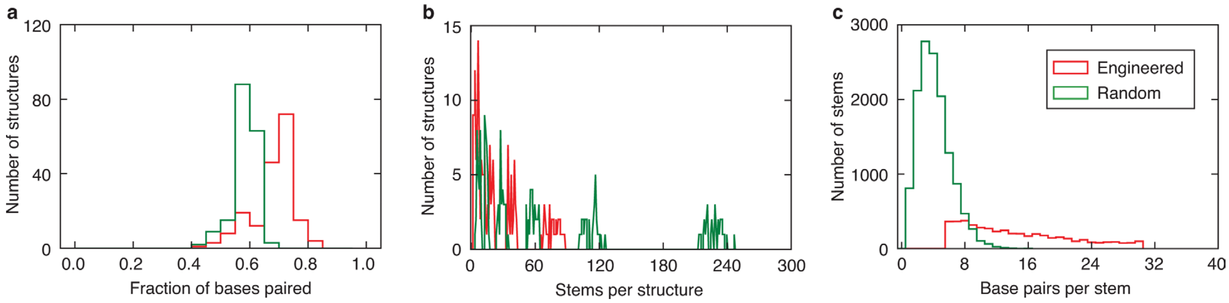

Figure 2. Comparison of the structural features of the engineered and random test sets.

Optimization is performed using a modified version of our algorithm with μ^k taking the role of n^k (pseudocode in Supporting Information Algorithm S5).

Implementation

The sequence design algorithm is coded in the C programming language. By parallelizing the dynamic program for evaluating $P(\phi)$ using MPI,[27] the sequence design algorithm is capable of exploiting multiple computational cores to reduce run time. For a design job allocated M computational cores, each evaluation of P^k for node k with structure s^k is performed using m cores for some $m \in 1, \ldots, M$ selected to approximately minimize run time based on $|s^k|$. Our sequence design algorithm is available online as part of the NUPACK web server (`www.nupack.org`).[28] NUPACK source code may be downloaded for non-commercial research purposes.

Results

Our primary test scenario is RNA sequence design at 37°C for target structures in the engineered test set using a single computational core. For each target structure in a test set, 10 independent design trials were performed. Each plotted data point represents a median over 300 design trials (10 trials for each of 30 structures for a given size N).

Algorithm Performance and Asymptotic Optimality

Figure 3 demonstrates the typical performance of our algorithm across a range of values of N using the engineered and random test sets. Typical designs surpass the desired design quality ($n(\phi, s) \leq N/100$) as a result of overshooting stop conditions lower in the decomposition tree (panel a). For the engineered test set, typical design cost ranges from a fraction of a second for $N = 100$ to roughly 3 h for $N = 3200$ (panel b). For small N, the design cost for the random test set is higher than for the engineered test set, becoming comparable as N increases. Typical `GC` content is less than 60% (starting from random initial sequences with ≈50% `GC` content; panel c). Remarkably, as the depth of the decomposition tree increases with N, the relative cost of design, $c_{\text{des}}(N)/c_{\text{eval}}(N)$, decreases asymptotically to the optimal bound of 4/3 (panel d). Hence, for sufficiently large N, the typical cost of sequence design is only 4/3 the cost of a single mutation evaluation on the root node. Mutation evaluation has time complexity $\Theta(N^3)$ and is empirically observed to be approximately in the asymptotic regime (Supporting Information Fig. S2). Hence, for our design algorithm, the empirical observation of asymptotic optimality implies that the exponent in the $\Omega(N^3)$ time complexity bound is sharp.

Leaf Independence and Emergent Defects

Figure 4 compares the ensemble defect evaluated at the root node, to the sum of the ensemble defects evaluated at the leaf nodes.[†] If the assumption of leaf independence is valid (i.e., if dummy nucleotides do a good job of mimicking parental environments and there is minimal crosstalk between merged subsequences), we would expect the data to fall near the diagonal.

For the engineered test set (panel a), we observe three striking properties. First, for random initial sequences, the assumption of leaf independence is well-justified despite the fact that the ensemble defect is large. Second, leaf optimization followed by merging without reoptimization (i.e., $M_{\text{reopt}} = 0$) typically yields full sequence designs that achieve the desired design quality ($n(\phi, s) \leq N/100$ on the root), with emergent defects arising only in a minority of cases. Third, these emergent defects are successfully eliminated by defect-weighted child sampling and reoptimization starting from new random initial subsequences. The resulting full sequence designs exhibit leaf independence and satisfy the stop condition.

By comparison, for the random test set, merging of leaf-optimized sequences typically does lead to emergent defects in the root node. Even in this case, our algorithm successfully eliminates emergent defects using defect-weighted child sampling and reoptimization starting from new random initial subsequences.

Contributions of Algorithmic Ingredients

Figure 5 isolates the contributions of hierarchical structure decomposition and defect-weighted sampling to our ensemble defect optimization algorithm by comparing performance to three modified algorithms lacking one or both ingredients. All four methods typically achieve the desired design quality, with hierarchical methods surpassing the quality requirement for the root node as a result of overshooting stop conditions lower in the decomposition tree. Hierarchical methods dramatically reduce design cost relative to their single-scale counterparts (which are not tested for $N = 800$ due to high cost). Defect-weighted sampling reduces design cost and `GC` content by focusing mutation effort on the most defective subsequences. For the single-scale methods, the relative cost of design, $c_{\text{des}}(N)/c_{\text{eval}}(N)$, increases with N. For hierarchical methods, $c_{\text{des}}(N)/c_{\text{eval}}(N)$ decreases asymptotically to the optimal bound of 4/3 as N increases. Our algorithm thus combines the design quality of ensemble defect optimization, the reduced cost and asymptotic optimality of hierarchical decomposition, and the reduced cost and reduced `GC` content of defect-weighted sampling.

Sequence Initialization

To explore the effect of sequence initialization on typical design quality and cost, we tested four types of initial conditions (Fig. 6): random sequences (default), random sequences using only `A` and `T` bases, random sequences using only `G` and `C` bases, and sequences satisfying sequence symmetry minimization (SSM).[29‡] The desired design quality is achieved independent of the initial conditions (panel a), which have little effect on design cost (panels b and d). Designs initiated with random `AT` sequences or with random `GC` sequences illustrate that the ensemble defect stop condition can be satisfied over a broad range of `GC` contents (panel c).

[†] To avoid overcounting defects at the leaves, n_i^k is counted in leaf k only if nucleotide i is native throughout its ancestry.

[‡] SSM is a heuristic that promotes specificity for the target structure by prohibiting repeated subsequences of a specified word length (taken to be six for our tests). For bases in single-stranded or branched regions of the target structure, the complementary word is also prohibited.[29]

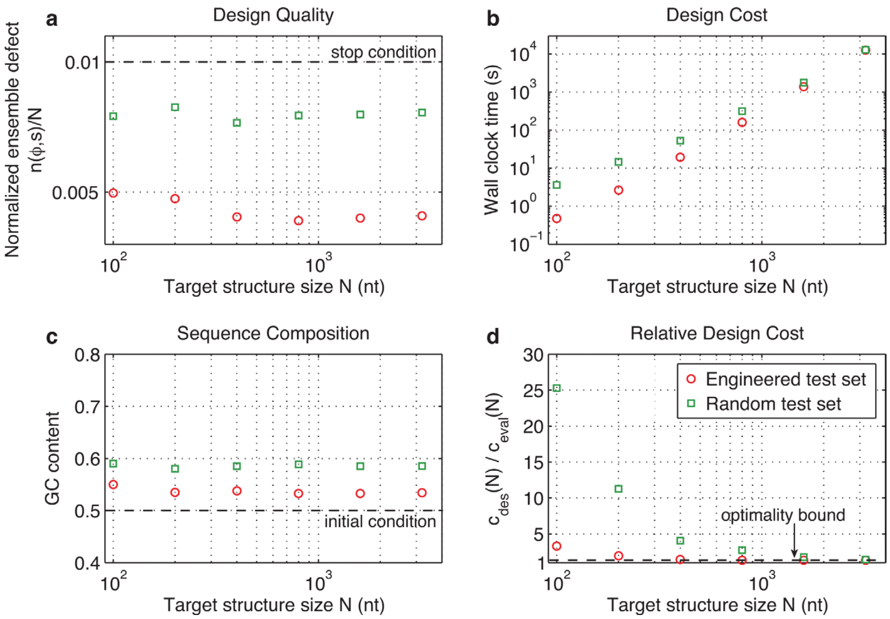

Figure 3. Algorithm performance and asymptotic optimality. a) Design quality. The stop condition is depicted as a dashed line. b) Design cost. c) Sequence composition. The initial GC content is depicted as a dashed line. d) Cost of sequence design relative to a single evaluation of the objective function. The optimality bound is depicted as a dashed line. RNA design at 37°C on the engineered and random test sets.

Stop Condition Stringency

Figure 7 depicts typical algorithm performance for five different levels of stringency in the stop condition: $f_{\text{stop}} \in \{0.001, 0.005, 0.01\,(\text{default}), 0.05, 0.10\}$. For each stop condition, the observed design quality is better than required as a result of overshooting stop conditions lower in the decomposition tree. Consistent with empirical asymptotic optimality, the design cost is independent of f_{stop} for sufficiently large N (for the tested stringency levels). It is noteworthy that the algorithm is capable of routinely and efficiently designing sequences with ensemble defect less than $N/1000$.

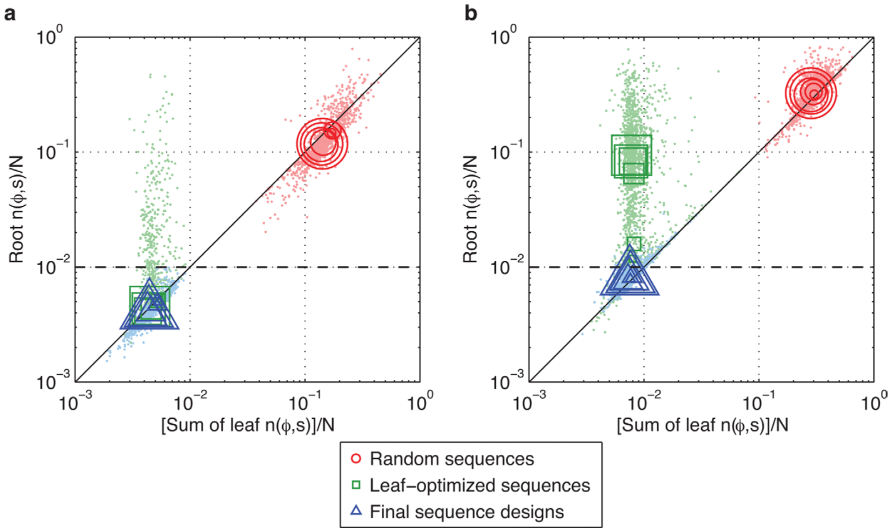

Figure 4. Leaf independence and emergent defects. Comparison of the ensemble defect evaluated at the root node to the sum of the ensemble defects evaluated at the leaf nodes. a) Engineered test set. b) Random test set. Dots represent independent designs. Symbols denote medians for each value of $N \in \{100, 200, 400, 800, 1600, 3200\}$ (symbol size increases with N). RNA design at 37°C.

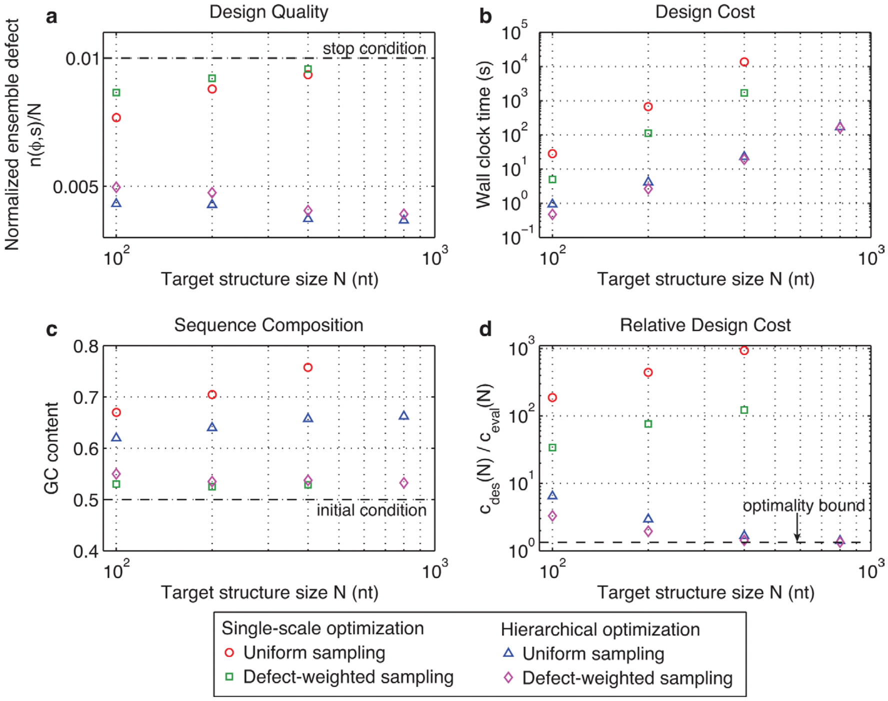

Figure 5. Contributions of hierarchical structure decomposition and defect-weighted sampling to algorithm performance. a) Design quality. The stop condition is depicted as a dashed line. b) Design cost. c) Sequence composition. The initial GC content is depicted as a dashed line. d) Cost of sequence design relative to a single evaluation of the objective function. The optimality bound is depicted as a dashed line. RNA design at 37°C on the engineered test set.

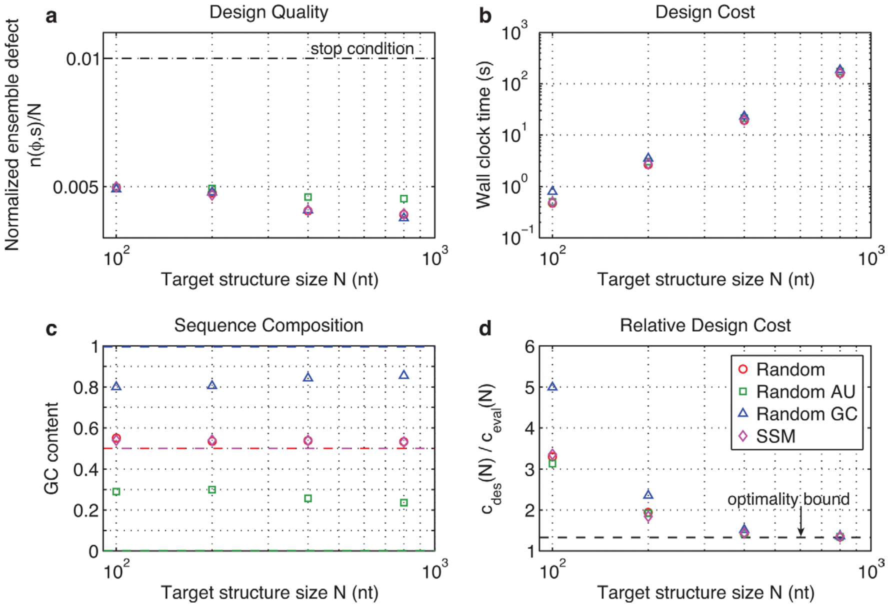

Figure 6. Effect of sequence initialization on algorithm performance. a) Design quality. The stop condition is depicted as a dashed line. b) Design cost. c) Sequence composition. Initial GC contents are depicted with dashed lines. d) Cost of sequence design relative to a single evaluation of the objective function. The optimality bound is depicted as a dashed line. RNA design at 37°C on the engineered test set.

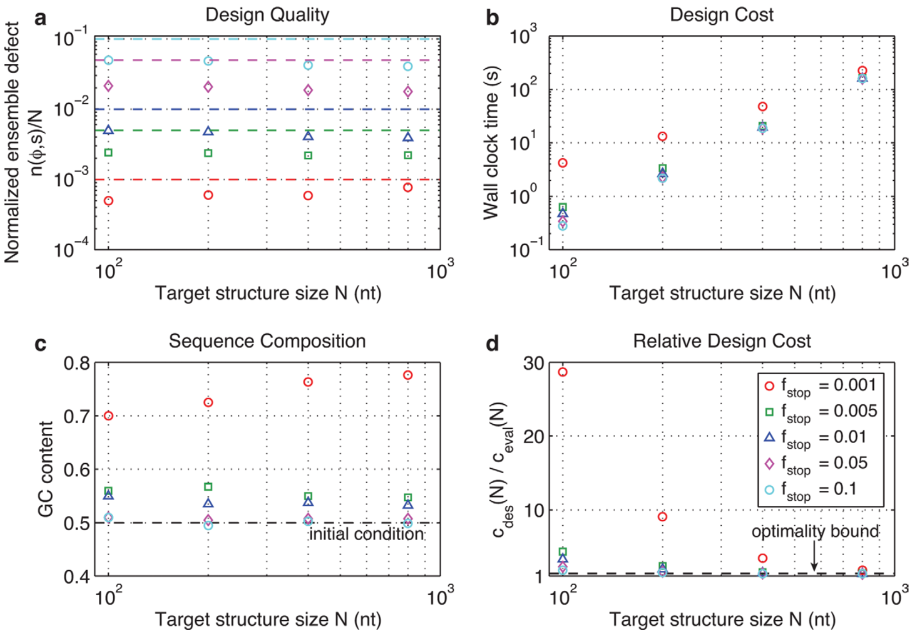

Figure 7. Effect of stop condition stringency on algorithm performance. a) Design quality. Stop conditions are depicted by dashed lines. b) Design cost. c) Sequence composition. The initial GC content is depicted as a dashed line. d) Cost of sequence design relative to a single evaluation of the objective function. The optimality bound is depicted as a dashed line. RNA design at 37°C on the engineered test set.

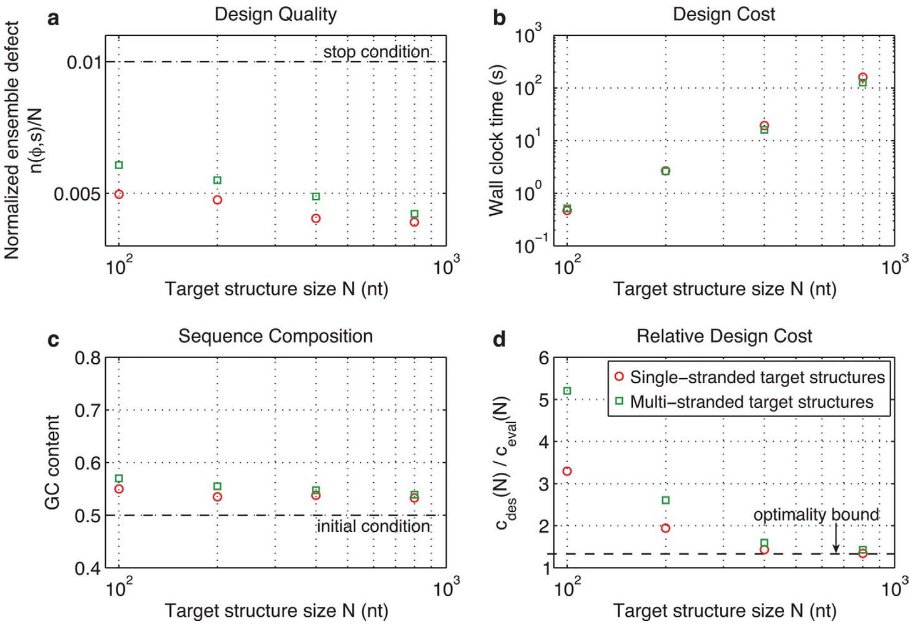

Figure 8. Algorithm performance on single-stranded and multi-stranded target structures. a) Design quality. The stop condition is depicted as a dashed line. b) Design cost. c) Sequence composition. The initial GC content is depicted as a dashed line. d) Cost of sequence design relative to a single evaluation of the objective function. The optimality bound is depicted as a dashed line. RNA design at 37°C on the engineered test set.

Multi-Stranded Target Structures

Multi-stranded target structures arise frequently in engineering practice.[1–3] Figure 8 demonstrates that our algorithm performs similarly on single-stranded and multi-stranded target structures.

Design Material

Figure 9 compares RNA and DNA design. DNA designs are performed in 1 M Na^+ at 23°C to reflect that DNA systems are typically engineered for room temperature studies. In comparison to RNA design, DNA design leads to similar design quality (panel a), higher design cost (panel b), and somewhat higher GC content (panel c), while continuing to exhibit asymptotic optimality (panel d).

Sequence Constraints and Pattern Prevention

Molecular engineers sometimes constrain the sequence of certain nucleotides in the target structure (e.g., to ensure complementarity to a specific biological sequence), or prevent certain patterns from appearing anywhere in the design (e.g., GGGG). Our algorithm accepts sequence constraints and pattern prevention requirements expressed using standard nucleic acid codes.[§] Figure 10 demonstrates that the prevention of patterns {AAAA, CCCC, GGGG, UUUU, KKKKKK, MMMMMM, RRRRRR, SSSSSS, WWWWWW, YYYYYY} has little effect on design quality or GC content (panels a and c), and somewhat increases design cost while retaining asymptotic optimality (panels b and d).

Parallel Efficiency and Speedup

The contour plots of Figure 11 demonstrate the parallel efficiency and speedup achieved using a parallel implementation of the design algorithm on M computational cores (efficiency$(N,M) = t(N,1)/(t(N,M) \times M)$, speedup$(N,M) = t(N,1)/t(N,M)$, where t is wall clock time). Using two computational cores, the parallel efficiency exceeds ≈0.9 for target structures with $N > 400$. Using 32 computational cores, the parallel speedup is ≈14 for target structures with $N = 3200$.

Comparison to Previous Approaches

Figure 12 compares the performance of our algorithm to the performance of algorithms inspired by previous publications. Single-scale methods that employ uniform mutation sampling to optimize either ensemble defect or probability defect achieve the desired design quality at significantly higher cost and with significantly higher GC content (panels a-c). Sequences resulting from probability defect optimization typically surpass the ensemble defect stop condition despite failing to satisfy the probability defect stop condition (panel e), reflecting the pessimism of $\pi(\phi, s)$ in characterizing the equilibrium structural defect over ensemble Γ. For either single-scale method, the relative cost of design, $c_{\mathrm{des}}(N)/c_{\mathrm{eval}}(N)$, increases with N (panel d). Owing to the high cost of the single-scale approaches, designs were not attempted for large N.

By contrast, hierarchical MFE defect optimization with defect-weighted sampling leads to efficient satisfaction of the MFE stop condition (panels b and f), exhibiting asymptotic optimality with $c_{\mathrm{des}}(N)/c_{\mathrm{eval}}(N)$ approaching 4/3 for large N (panel d). Asymptotically, the cost of hierarchical MFE defect optimization relative to hierarchical ensemble defect optimization is lower by a constant factor corresponding to the relative cost of evaluating the two objective functions using $\Theta(N^3)$ dynamic programs (panels b and d). The shortcoming of MFE defect optimization is the unreliability of $s^{\mathrm{MFE}}(\phi)$ in characterizing the equilibrium structural properties of ensemble Γ.[24] Despite satisfying the MFE defect stop condition, sequences designed via MFE defect optimization typically fail to achieve the ensemble defect stop condition by roughly a factor of five for the engineered test set (panel a), and by roughly a factor of 20 for the random test set (Supporting Information Fig. S3).

Discussion

Our algorithm combines four major ingredients to design the sequence ϕ of one or more strands intended to adopt target secondary structure s at equilibrium:

- *Ensemble defect optimization*: The design objective function is the ensemble defect, $n(\phi, s)$, representing the average number of incorrectly paired nucleotides at equilibrium calculated over the ensemble of unpseudoknotted secondary structures Γ. For a target structure with N nucleotides, we seek to satisfy the stop condition: $n(\phi, s) \leq f_{\mathrm{stop}}N$.
- *Hierarchical structure decomposition*: We perform a binary tree decomposition of the target secondary structure, decomposing each parent structure within a duplex stem, and introducing dummy nucleotides to extend the truncated duplex in each child structure to mimic the parental environment.
- *Leaf optimization with defect-weighted mutation sampling*: Starting from a random initial sequence, sequence optimization is performed in the leaf nodes using defect-weighted mutation sampling in which each candidate mutation position is selected with probability proportional to its contribution to the ensemble defect of the leaf.
- *Subsequence merging and reoptimization*: As subsequences are merged moving up the tree, a parent node initiates defect-weighted child sampling and reoptimization within its subtree only if there are emergent defects resulting from crosstalk between child subsequences. Leaf reoptimization starts from a new random initial sequence.

Using a $\Theta(N^3)$ dynamic program to evaluate the design objective function, we derive an asymptotic optimality bound on design time: for large N, the minimum cost to design a sequence with N nucleotides is 4/3 the cost of evaluating the objective function once on N nucleotides. Hence, our design algorithm has time complexity $\Omega(N^3)$.

[§] During leaf optimization, mutation candidates are not considered if they would introduce a pattern violation. Pattern violations that arise during merging are eliminated via an adaptive walk in which mutations are accepted if they reduce the number of pattern violations.

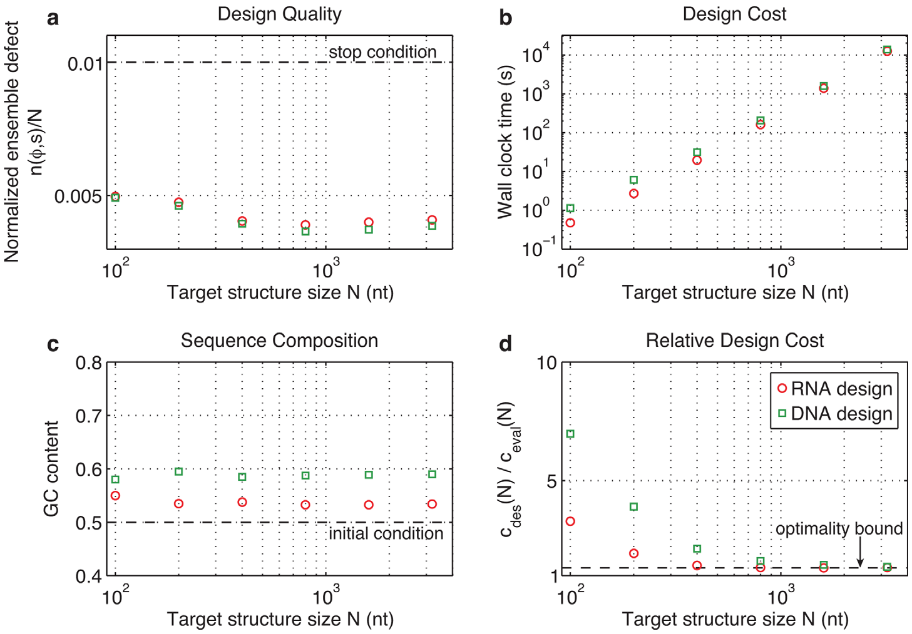

Figure 9. Effect of design material on algorithm performance. a) Design quality. The stop condition is depicted as a dashed line. b) Design cost. c) Sequence composition. The initial GC content is depicted as a dashed line. d) Cost of sequence design relative to a single evaluation of the objective function. The optimality bound is depicted as a dashed line. RNA design at 37°C and DNA design at 23° on the engineered test set.

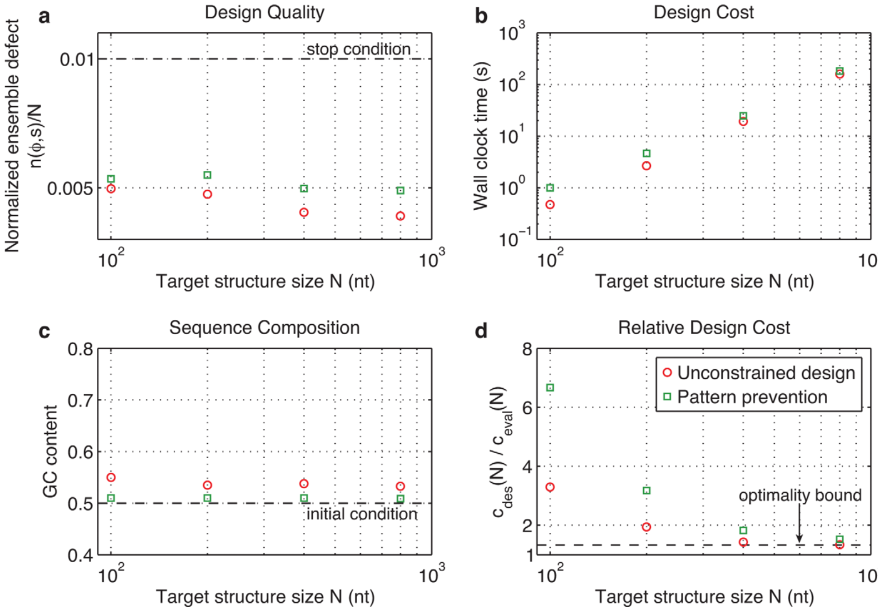

Figure 10. Effect of pattern prevention on algorithm performance. a) Design quality. The stop condition is depicted as a dashed line. b) Design cost. c) Sequence composition. The initial GC content is depicted as a dashed line. d) Cost of sequence design relative to a single evaluation of the objective function. The optimality bound is depicted as a dashed line. RNA design at 37°C on the engineered test set.

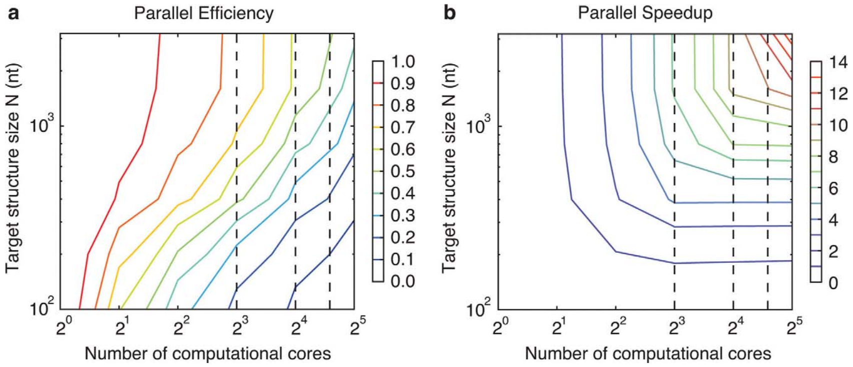

Figure 11. Parallel algorithm performance. a) Parallel efficiency and b) parallel speedup using multiple computational cores. Dashed lines denote boundaries between nodes, indicating the use of message passing. RNA design at 37°C on the engineered test set.

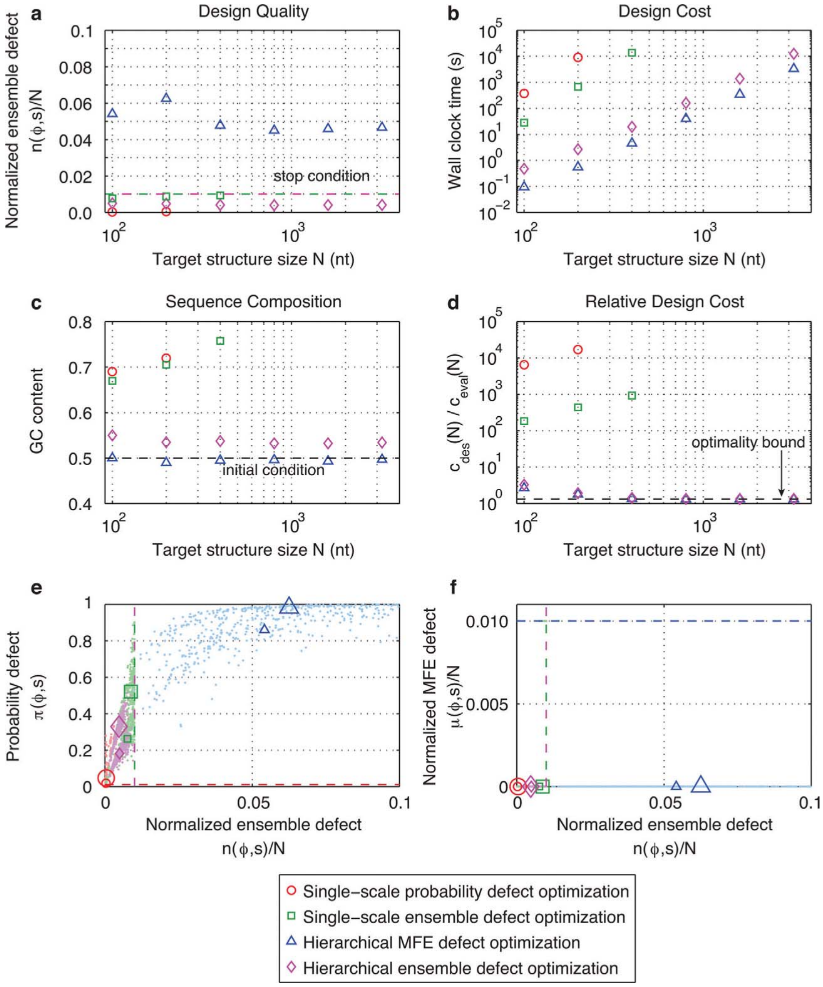

Figure 12. Comparison to algorithms inspired by previous publications. a) Design quality. The stop condition for ensemble defect optimization is depicted as a dashed line. b) Design cost. c) Sequence composition. The initial GC content is depicted as a dashed line. d) Cost of sequence design relative to a single evaluation of the objective function. The optimality bound is depicted as a dashed line. e,f) Evaluation of each sequence design using three objective functions. Stop conditions are depicted as dashed lines. Dots represent independent designs. Symbols denote medians for each value of $N \in \{100, 200\}$ (symbol size increases with N). RNA design at 37°C on the engineered test set.

We studied the performance of our algorithm in the context of empirical secondary structure free energy models[6,7] that have practical utility for the analysis[30–34] and design[35–39] of functional nucleic acid systems. In particular, we examined RNA design at 37°C on target structures containing $N \in \{100, 200, 400, 800, 1600, 3200\}$ nucleotides and duplex stems ranging from 1 to 30 base pairs. Empirically, we observe several striking properties:

- Emergent defects are sufficiently infrequent that they can typically be eliminated by leaf reoptimization starting from new random initial sequences.
- It is routine to design sequences with ensemble defect $n(\phi, s) < N/100$ over a wide range of GC contents.
- Our algorithm exhibits asymptotic optimality for large N, with full sequence design costing roughly 4/3 the cost of a single evaluation of the objective function. Hence, the algorithm is efficient in the sense that the exponent in the $\Omega(N^3)$ time complexity bound is sharp.

We modified our algorithm to compare performance to algorithms inspired by previous work.[14,20–22,24–26] In line with conceptual expectations, we observe empirically that our algorithm achieves lower design cost relative to single-scale probability or ensemble defect optimization with uniform mutation sampling, and higher design quality relative to hierarchical MFE defect optimization with defect-weighted sampling.

To enhance the utility of our algorithm for molecular engineers, our algorithm addresses several practical considerations, including: sequence constraints, pattern prevention, multi-stranded target structures, and parallel execution.

Acknowledgments

The authors thank R.M. Dirks and L.B. Pierce for helpful discussions.

Appendix

Algorithm 1. Pseudocode for hierarchical ensemble defect optimization with defect-weighted sampling. For a given target structure s, a designed sequence ϕ is returned by the function call DESIGNSEQ($\emptyset, s, \emptyset, 1$). During the recursive design procedure, ϕ, s, and n are local variables that are used to push sequence, structure, and defect information between nodes in the tree. By contrast, $n^{k,a}$ provides global storage for the ensemble defect of each node k. For a given k, the index, $a = 1, \ldots,$DEPTH(k), enables storage of the ensemble defect corresponding to the sequence for node k that has been accepted up to depth a in the tree. Storage of these historical values eliminates unnecessary recalculation of ensemble defects during subtree reoptimization.

```
DESIGNSEQ($\phi, s, n, k$)
    $a \leftarrow$ DEPTH($k$)
    if HASCHILDREN($k$)
        $m_{\text{reopt}} \leftarrow 0$
        if $n = \emptyset$
            $\phi_l \leftarrow$ DESIGNSEQ($\emptyset, s_{l+}, \emptyset, k_l$)
            $\phi_r \leftarrow$ DESIGNSEQ($\emptyset, s_{r+}, \emptyset, k_r$)
        else
            UPDATECHILDREN($k, a, a-1$)
            child, $\phi \leftarrow$ WEIGHTEDCHILDSAMPLING($\phi, s, n_l, n_r$)
            $\phi_{\text{child}} \leftarrow$ DESIGNSEQ($\phi_{\text{child}+}, s_{\text{child}+}, n_{\text{child}+}, k_{\text{child}}$)
        $n^{k,a} \leftarrow$ ENSEMBLEDEFECT($\phi, s$)
        UPDATECHILDREN($k, a, a+1$)
        while $n^{k,a} > \max(f_{\text{stop}}|s_l|, n^{k_l,a}_{\text{native}}) + \max(f_{\text{stop}}|s_r|, n^{k_r,a}_{\text{native}})$
                and $m_{\text{reopt}} < M_{\text{reopt}}$
            child, $\hat{\phi} \leftarrow$ WEIGHTEDCHILDSAMPLING($\phi, s, n^{k,a}_l, n^{k,a}_r$)
            $\hat{\phi}_{\text{child}} \leftarrow$ DESIGNSEQ($\phi_{\text{child}+}, s_{\text{child}+}, n^{k,a}_{\text{child}+}, k_{\text{child}}$)
            $\hat{n} \leftarrow$ ENSEMBLEDEFECT($\hat{\phi}, s$)
            if $\hat{n} < n^{k,a}$
                $\phi, n^{k,a} \leftarrow \hat{\phi}, \hat{n}$
                UPDATECHILDREN($k, a, a+1$)
            $m_{\text{reopt}} \leftarrow m_{\text{reopt}} + 1$
    else
        $m_{\text{leafopt}} \leftarrow 0$
        $\phi, n^{k,a} \leftarrow$ OPTIMIZELEAF($s$)
        while $n^{k,a} > f_{\text{stop}}|s|$ and $m_{\text{leafopt}} < M_{\text{leafopt}}$
            $\hat{\phi}, \hat{n} \leftarrow$ OPTIMIZELEAF($s$)
            if $\hat{n} < n^{k,a}$
                $\phi, n^{k,a} \leftarrow \hat{\phi}, \hat{n}$
            $m_{\text{leafopt}} \leftarrow m_{\text{leafopt}} + 1$
    return $\phi_{\text{native}}$
```

```
UPDATECHILDREN($k, a, b$)
    if HASCHILDREN($k$)
        $n^{k_l,a} \leftarrow n^{k_l,b}$
        $n^{k_r,a} \leftarrow n^{k_r,b}$
        UPDATECHILDREN($k_l, a, b$)
        UPDATECHILDREN($k_r, a, b$)

OPTIMIZELEAF($s$)
    $m_{\text{unfavorable}} \leftarrow 0$
    $\gamma_{\text{unfavorable}} \leftarrow \emptyset$
    $\phi \leftarrow$ INITSEQ($s$)
    $n \leftarrow$ ENSEMBLEDEFECT($\phi, s$)
    while $n > f_{\text{stop}}|s|$ and $m_{\text{unfavorable}} < M_{\text{unfavorable}}|s|$
        $\xi, \hat{\phi} \leftarrow$ WEIGHTEDMUTATIONSAMPLING($\phi, s, n_1, \ldots, n_{|s|}$)
        if $\xi \in \gamma_{\text{unfavorable}}$
            $m_{\text{unfavorable}} \leftarrow m_{\text{unfavorable}} + 1$
        else
            $\hat{n} \leftarrow$ ENSEMBLEDEFECT($\hat{\phi}, s$)
            if $\hat{n} < n$
                $\phi, n \leftarrow \hat{\phi}, \hat{n}$
                $m_{\text{unfavorable}} \leftarrow 0$
                $\gamma_{\text{unfavorable}} \leftarrow \emptyset$
            else
                $m_{\text{unfavorable}} \leftarrow m_{\text{unfavorable}} + 1$
                $\gamma_{\text{unfavorable}} \leftarrow \gamma_{\text{unfavorable}} \bigcup \xi$
    return $\phi, n$
```

References

1. Simmel, F. C.; Dittmer, W. U. Small 2005, 1, 284.
2. Feldkamp, U.; Niemeyer, C. M. Angew Chem Int Ed Engl 2006, 45, 1856.
3. Bath, J.; Turberfield, A. J. Nat Nanotechnol 2007, 2, 275.
4. Dirks, R. M.; Bois, J. S.; Schaeffer, J. M.; Winfree, E.; Pierce, N. A. SIAM Rev 2007, 49, 65.
5. Serra, M. J.; Turner, D. H. Methods Enzymol 1995, 259, 242.
6. Mathews, D. H.; Sabina, J.; Zuker, M.; Turner, D. H. J Mol Biol 1999, 288, 911.
7. SantaLucia, J., Jr. Proc Natl Acad Sci USA 1998, 95, 1460.
8. SantaLucia, J.; Hicks, D. Annu Rev Biophys Biomol 2004, 33, 415.
9. Koehler, R. T.; Peyret, N. Bioinformatics 2005, 21, 3333.
10. Landau, L. D.; Lifshitz, E. M. Statistical Physics Part 1, 3rd ed., Butterworth-Heinemann: New York, 1980.
11. Zuker, M.; Sankoff, D. Bull Math Biol 1984, 46, 591.
12. Zuker, M.; Stiegler, P. Nucleic Acids Res 1981, 9, 133.
13. McCaskill, J. S. Biopolymers 1990, 29, 1105.
14. Hofacker, I. L.; Fontana, W.; Stadler, P. F.; Bonhoeffer, L. S.; Tacker, M.; Schuster, P. Chem Mon 1994, 125, 167.
15. Lyngsø, R. B.; Zuker, M.; Pedersen, C. N. S. Bioinformatics 1999, 15, 440.
16. Dirks, R. M.; Pierce, N. A. J Comput Chem 2004, 25, 1295.
17. Dimitrov, R. A. Zuker, M. Biophys J 2004, 87, 215.
18. Andronescu, M.; Zhang, Z. C.; Condon, A. J Mol Biol 2005, 345, 987.
19. Bernhart, S. H.; Tafer, H.; Muckstein, U.; Flamm, C.; Stadler, P. F.; Hofacker, I. L. Algorithms Mol Biol 2006, 1, 3.
20. Andronescu, M.; Fejes, A. P.; Hutter, F.; Hoos, H. H.; Condon, A. J Mol Biol 2004, 336, 607.
21. Busch, A.; Backofen, R. Bioinformatics 2006, 22, 1823.
22. Aguirre-Hernandez, R.; Hoos, H. H.; Condon, A. BMC Bioinformatics 2007, 8, 34.
23. Burghardt, B.; Hartmann, A. K. Phys Rev E 2007, 75, 021920.
24. Dirks, R. M.; Lin, M.; Winfree, E.; Pierce, N. A. Nucleic Acids Res 2004, 32, 1392.
25. Flamm, C.; Hofacker, I. L.; Maurer-Stroh, S.; Stadler, P. F.; Zehl, M. RNA 2001, 7, 254.
26. Dirks, R. M.; Pierce, N. A. J Comput Chem 2003, 24, 1664.
27. Fekete, M.; Hofacker, I. L.; Stadler, P. F. J Comput Biol 2000, 7, 171.
28. Zadeh, J. N.; Steenberg, C. D.; Bois, J. S.; Wolfe, B. R.; Khan, A. R.; Pierce, M. B.; Dirks, R. M.; Pierce, N. A. J Comput Chem, in press.
29. Seeman, N. C. J Theor Biol 1982, 99, 237.
30. Ding, Y.; Lawrence, C. E. Nucleic Acids Res 2003, 31, 7280.
31. Mathews, D. H. RNA 2004, 10, 1178.
32. Ding, Y.; Chan, C. Y.; Lawrence, C. E. RNA 2005, 11, 1157.
33. Rogic, S.; Montpetit, B.; Hoos, H. H.; Mackworth, A. K.; Ouellette, B. F.; F.; Hieter, P. BMC Genomics 2008, 9, 355.
34. Zhi, J. L.; Gloor, J. W.; Mathews, D. H. RNA 2009, 15, 1805.
35. Dirks, R. M.; Pierce, N. A. Proc Natl Acad Sci USA 2004, 101, 15275.
36. Patzel, V.; Rutz, S.; Dietrich, I.; Köberle, C.; Sheffold, A.; Kaufmann, S. H. E. Nat Biotechnol 2005, 23, 1440.
37. Penchovsky, R.; Breaker, R. R. Nat Biotechnol 2005, 23, 1424.
38. Venkataraman, S.; Dirks, R. M.; Rothemund, P. W. K.; Winfree, E.; Pierce, N. A. Nat Nanotechnol 2007, 2, 490.
39. Yin, P.; Choi, H. M. T.; Calvert, C. R.; Pierce, N. A. Nature 2008, 451, 318.