

NUPACK: Analysis and Design of Nucleic Acid Structures, Devices, and Systems

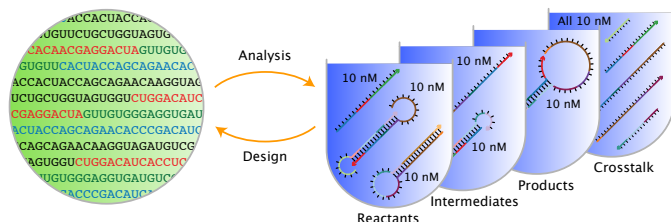
Mark E. Fornace,^{1,2,#} Jining Huang^{1,#}, Cody T. Newman^{1,#}, Nicholas J. Porubsky², Marshall B. Pierce,
and Niles A. Pierce^{1,3,*}

¹Division of Biology & Biological Engineering, California Institute of Technology, Pasadena, CA 91125, USA. ²Division of Chemistry & Chemical Engineering, California Institute of Technology, Pasadena, CA 91125, USA. ³Division of Engineering & Applied Science, California Institute of Technology, Pasadena, CA 91125, USA.

ABSTRACT:

NUPACK is a growing software suite for the analysis and design of nucleic acid structures, devices, and systems serving the needs of researchers in the fields of nucleic acid nanotechnology, molecular programming, synthetic biology, and across the life sciences. NUPACK algorithms are unique in treating complex and test tube ensembles containing arbitrary numbers of interacting strand species, providing crucial tools for capturing concentration effects essential to analyzing and designing the intermolecular interactions that are a hallmark of these fields. The all-new NUPACK web app (nupack.org) has been re-architected for the cloud, leveraging a cluster that scales dynamically in response to user demand to enable rapid job submission and result inspection even at times of peak user demand. The web app exploits the all-new NUPACK 4 scientific code base as its backend, offering enhanced physical models (coaxial and dangle stacking subensembles), dramatic speedups (20–120× for test tube analysis), and increased scalability for large complexes. NUPACK 4 algorithms can also be run locally using the all-new NUPACK Python module.

KEYWORDS: DNA, RNA, secondary structure, base-pairing, hybridization, complex ensemble, test tube ensemble, multi-tube ensemble, equilibrium, partition function, concentration, ensemble defect, analysis, design, reaction pathway engineering



INTRODUCTION

We are engaged in a multi-decade effort to develop NUPACK (Nucleic Acid Package), a growing software suite for the analysis and design of nucleic acid structures, devices, and systems.¹ NUPACK algorithms^{2–9} are formulated in terms of nucleic acid secondary structure (i.e., the base-pairs of a set of DNA or RNA strands) and employ empirical free energy parameters.^{10–22}

Problem categories. NUPACK algorithms address two fundamental classes of problems:

- *Sequence analysis:* given a set of DNA or RNA strands, analyze the equilibrium base-pairing properties over a specified ensemble.
- *Sequence design:* given a set of desired equilibrium base-pairing properties, design the sequences of a set of DNA or RNA strands over a specified ensemble. Sequence design is performed subject to diverse sequence constraints.

Ensembles. NUPACK algorithms operate over two fundamental ensembles:

- *Complex ensemble:* the ensemble of all (unpseudoknotted connected) secondary structures for an arbitrary number of interacting RNA or DNA strands.
- *Test tube ensemble:* the ensemble of a dilute solution containing an arbitrary number of RNA or DNA strand species (introduced at user-specified concentrations) interacting to form an arbitrary number of complex species.

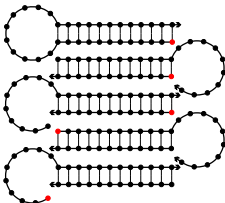
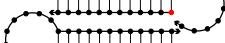
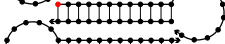
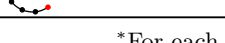

Furthermore, to enable reaction pathway engineering of dynamic hybridization cascades (e.g., shape and sequence transduction using small conditional RNAs^{23,24}) and large-scale structural engineering including pseudoknots (e.g., RNA origamis²⁵), NUPACK generalizes sequence analysis and design to *multi-tube ensembles* comprising one or more test tubes.⁸

Note that a complex ensemble is subsidiary to a test tube ensemble, so complex analysis is inherent in test tube analysis (but not vice versa), and complex design is inherent in test tube design (but not vice versa). As it is typically infeasible to experimentally study a single complex in isolation, we recommend analyzing and designing nucleic acid strands in a test tube ensemble that contains the complex of interest as well as other competing complexes that might form in solution. For example, if one is experimentally studying strands A and B that are intended to predominantly form a secondary structure within the ensemble of complex A·B, one should not presuppose that the strands do indeed form A·B and simply analyze or design the base-pairing properties of that complex. Instead, it is more physically relevant to analyze or design a test tube ensemble containing strands A and B interacting to form multiple complex species (e.g., A, B, A·A, A·B, B·B) so as to capture both concentration information (how much A·B forms?) and structural information (what are the base-pairing properties of A·B when it does form?).

All-new NUPACK 4 scientific code base. NUPACK 4 analysis algorithms employ a new unified dynamic programming framework⁹ that provides enhanced secondary structure models including coaxial and dangle stacking subensembles, dramatic speedups (e.g., 20–120× for analysis of test tube ensembles), and enhanced scalability for large complexes (e.g., 30,000 nt). NUPACK 4 design algorithms leverage these performance benefits and support both hard constraints (that must be obeyed by the designed sequences; e.g., diversity and biological sequence constraints) and soft constraints (that penalize but do not prohibit suboptimal sequences; e.g., sequence symmetry and energy match constraints) for multi-tube ensembles.

All-new cloud-based NUPACK web app. Since its launch in 2007, usage of the NUPACK web app (nupack.org)¹ has increased to the point where the underlying static compute cluster is frequently overwhelmed by user demand. To pro-

Table 1. Secondary structure notation: dot-parens plus, run-length encoded (RLE) dot-parens-plus, and DU+.

Secondary structure*	Dot-parens-plus	RLE dot-parens-plus	DU+
	(((((.....))))))	(12.10)12	D12 U10
	(((((+)))))).....	(12+)12.10	D12 + U10
	(((((+.....))))))	(12+.10)12	D12 (+ U10)
	(((((.....+))))))	(12.10+)12	D12 (U10 +)
(((+))))))	.10(12+)12	U10 D12 +

*For each secondary structure, the first nucleotide is depicted in red.

vide a scalable resource for the global research community, the NUPACK web app was re-architected from the ground up to exploit a scalable compute cluster that resizes dynamically in the cloud in response to user demand. The all-new NUPACK web app integrates diverse components to create an intuitive and powerful analysis and design environment:

- *Algorithms*: mathematically rigorous, physically sound, computationally efficient scientific algorithms.²⁻⁹
- *Hardware*: a hybrid cloud compute cluster combining local hardware and scalable cloud hardware.
- *Interface*: an intuitive web interface for rapid job submission and result inspection.
- *Graphics*: publication-quality client-side graphics to enable straightforward interpretation of results, interactive data, and efficient preparation of talks and papers.

Researchers can run jobs and inspect results within the NUPACK web app, which leverages NUPACK 4 scientific code base as its backend.

All-new NUPACK Python module. Alternatively, researchers can script and run jobs locally using the all-new NUPACK 4 Python module, providing the flexibility to interact with the broader Python ecosystem.

Outline. To provide a foundation for describing the NUPACK web app, we begin by defining the physical model, relevant physical quantities, and the design formulation. For the convenience of the reader, definitions and descriptions are drawn from the original algorithms papers,^{3,5-9} but are here organized into a single unified presentation for easy perusal. We then summarize the features of the Analysis, Design, and Utilities pages of the NUPACK web app:

- *Analysis page*: Analyze the equilibrium base-pairing properties of one or more test tube ensembles (and subsidiary complex ensembles). These are the all-purpose sequence analysis tools.
- *Design page*: Design the sequences for one or more test tube ensembles (and subsidiary complex ensembles). These are the all-purpose sequence design tools.
- *Utilities page*: Analyze, design, or prepare figures for a single complex ensemble. These are quick tools applicable when your ensemble is a single complex.

PHYSICAL MODEL

Sequence. The *sequence*, ϕ , of one or more interacting RNA strands is specified as a list of bases $\phi^a \in \{A, C, G, U\}$ for $a = 1, \dots, |\phi|$. For DNA, $\phi^a \in \{A, C, G, T\}$. Nucleic acid sequences are listed 5' to 3'. For RNA calculations, T is automatically converted to U, and vice versa for DNA calculations. For sequence design, sequence constraints can be specified using IUPAC degenerate nucleotide codes (see Table 2).

Table 2. IUPAC degenerate nucleotide codes for RNA.

Code	Nucleotides*
M	A or C
R	A or G
W	A or U
S	C or G
Y	C or U
K	G or U
V	A, C, or G
H	A, C, or U
D	A, G, or U
B	C, G, or U
N	A, C, G, or U

*For DNA, T replaces U.

Secondary structure. A *secondary structure*, s , of one or more interacting RNA strands is defined by a set of base pairs, each a Watson–Crick pair (A·U or C·G) or a wobble pair (G·U) (e.g., see Figure 1a). For DNA, the corresponding Watson–Crick pairs are A·T or C·G and there are no wobble pairs (G·T is classified as a mismatch¹⁸). A *polymer graph* representation of 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 *unpseudoknotted* if there exists a strand ordering for which the polymer graph has no crossing lines, or *pseudoknotted* if all strand orderings contain crossing lines. A secondary structure is *connected* if no subset of the strands is free of the others.⁵

Secondary structures may be specified in one of three ways for NUPACK calculations (see Table 1 for examples):

- *dot-parens-plus notation*: each unpaired base is represented by a dot, each base pair by matching parentheses, and each nick between strands by a plus.¹
- *run-length encoded (RLE) dot-parens-plus notation*: as a shorthand for dot-parens-plus, any sequence of consecutive characters in dot-parens-plus may be replaced by the character followed by a number.⁹
- *DU+ notation*: Using DU+ notation, a duplex is denoted by D followed by the number of base pairs and an unpaired region is denoted by U followed by the number of unpaired nucleotides.²⁶ Each duplex is followed immediately by the substructure (specified in DU+ notation) that is “enclosed” by the duplex. If this substructure includes more than one element, parentheses are used to denote scope. A nick between strands is specified by a “+”.

In mathematical expressions, it is convenient to represent secondary structure s using a *structure matrix* $S(s)$ with entries $S^{a,b}(s) = 1$ if structure s contains base pair $a \cdot b$

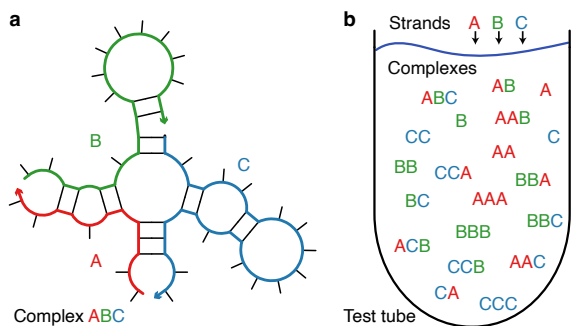


Figure 1. Complex and test tube ensembles. (a) A (connected unpsuedoknotted) secondary structure for a complex of 3 strands with strand ordering $\pi = ABC$. An arrowhead denotes the 3' end of each strand. (b) A test tube ensemble containing strand species $\Psi^0 = \{A,B,C\}$ interacting to form all complex species Ψ of up to $L_{\max} = 3$ strands. Adapted with permission from Fornace *et al.*, *ACS Synth Biol*, 9, 2665-2678, 2020. Copyright 2020 American Chemical Society.

and $S^{a,b}(s) = 0$ otherwise. Abusing notation, the entry $S^{a,a}(s) = 1$ if base a is unpaired in structure s and 0 otherwise. Hence, $S(s)$ is a symmetric matrix with row and column sums of 1.

Complex ensemble. Consider a complex of L distinct strands (each with a unique identifier in $\{1, \dots, L\}$) corresponding to strand ordering π . The *complex ensemble* $\bar{\Gamma}(\phi)$ contains all connected polymer graphs with no crossing lines (i.e., all unpsuedoknotted secondary structures).⁵ As a matter of algorithmic necessity, all of the dynamic programs in NUPACK operate on complex ensemble $\bar{\Gamma}(\phi)$ treating all strands as distinct. However, in the laboratory, strands with the same sequence are typically indistinguishable with respect to experimental observables. For comparison to experimental data, physical quantities calculated over ensemble $\bar{\Gamma}(\phi)$ are post-processed to obtain the corresponding quantities calculated over *complex ensemble* $\Gamma(\phi)$ in which strands with the same sequence are treated as indistinguishable.⁹ The ensemble $\Gamma(\phi) \subseteq \bar{\Gamma}(\phi)$ is a maximal subset of distinct secondary structures for strand ordering π . Two secondary structures are indistinguishable if their polymer graphs can be rotated so that all strands are mapped onto indistinguishable strands, all base pairs are mapped onto base pairs, and all unpaired bases are mapped onto unpaired bases; otherwise the structures are distinct.⁵

Test tube ensemble. A *test tube ensemble* is a dilute solution containing a set of strand species, Ψ^0 , introduced at user-specified concentrations, that interact to form a set of complex species, Ψ , each corresponding to a different strand ordering treating strands with the same sequence as indistinguishable.^{5,9} For L strands, there are $(L-1)!$ strand orderings if all strands are different species (e.g., complexes $\pi = ABC$ and $\pi = ACB$ for $L = 3$ and strands A, B, C), but fewer than $(L-1)!$ strand orderings if some strands are of the same species (e.g., complex $\pi = AAA$ for $L = 3$ with three A strands). By the Representation Theorem,⁵ a secondary structure in the complex ensemble for one strand ordering does not appear in the complex ensemble for any other strand ordering, averting redundancy. It is often convenient to define Ψ to contain all complex species of up to L_{\max} strands, although Ψ can be defined to contain arbitrary complex species formed from the strand species in Ψ^0 .

Multi-tube ensemble. Consider the multi-tube ensemble Ω where tube $h \in \Omega$ contains the set of strand species Ψ_h^0 interacting to form the set of complex species Ψ_h . The set of all complexes in multi-tube ensemble Ω is then $\Psi \equiv \cup_{h \in \Omega} \Psi_h$.

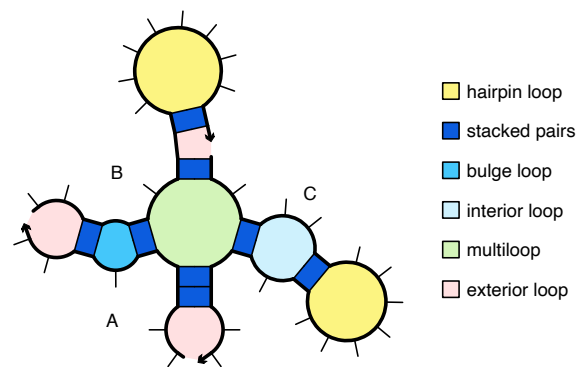


Figure 2. Loop-based free energy model for a complex. Canonical loop types for a complex with strand ordering $\pi = ABC$. Adapted with permission from Fornace *et al.*, *ACS Synth Biol*, 9, 2665-2678, 2020. Copyright 2020 American Chemical Society.

Note that the multi-tube ensemble encompasses the complex and test tube ensembles as subsidiary special cases.⁸

Loop-based free energy model. For each (unpsuedoknotted connected) secondary structure $s \in \bar{\Gamma}(\phi)$, the free energy, $\bar{\Delta G}(\phi, s)$, is estimated as the sum of the empirically determined free energies of the constituent loops^{12-14, 17, 21, 22} plus a strand association penalty,²⁷ ΔG^{assoc} , applied $L - 1$ times for a complex of L strands:

$$\bar{\Delta G}(\phi, s) = (L - 1) \Delta G^{\text{assoc}} + \sum_{\text{loop} \in s} \Delta G(\text{loop}). \quad (1)$$

The secondary structure of Figure 2 illustrates the different loop types, with loop free energy, $\Delta G(\text{loop})$, modeled as follows:^{12-14, 17, 21, 22}

- A *hairpin loop* is closed by a single base-pair $a \cdot b$. The loop free energy, $\Delta G_{a,b}^{\text{hairpin}}$, depends on sequence and loop size.
- An *interior loop* is closed by two base pairs ($a \cdot b$ and $d \cdot e$ with $a < d < e < b$). The loop free energy, $\Delta G_{a,d,e,b}^{\text{interior}}$ depends on sequence, loop size, and loop asymmetry. *Bulge loops* (where either $d = a + 1$ or $e = b - 1$) and *stacked pairs* (where both $d = a + 1$ and $e = b - 1$) are treated as special cases of interior loops.
- A *multiloop* is closed by three or more base pairs. The loop free energy is modeled as the sum of three sequence-independent penalties: $\Delta G_{\text{init}}^{\text{multi}}$ for formation of a multiloop, $\Delta G_{\text{bp}}^{\text{multi}}$ for each closing base pair, $\Delta G_{\text{nt}}^{\text{multi}}$ for each unpaired nucleotide inside the multiloop, one sequence-dependent penalty: $\Delta G_{a,b}^{\text{terminalbp}}$ for each closing pair $a \cdot b$, and optional coaxial and dangle stacking contributions (see below).
- An *exterior loop* contains a nick between strands and any number of closing base pairs. The exterior loop free energy is the sum of $\Delta G_{a,b}^{\text{terminalbp}}$ for each closing base pair $a \cdot b$, plus optional coaxial and dangle stacking contributions (see below). Hence, an unpaired strand has a free energy of zero, corresponding to the reference state.⁵

See Section S1.7 of Reference 9 for details on the functional form of loop-based free energy models.

Coaxial and dangle stacking. Within a multiloop or an exterior loop, there is a subensemble of coaxial stacking states between adjacent closing base pairs and dangle stacking states between closing base pairs and adjacent unpaired bases. Within a multiloop or exterior loop, a base pair can form one *coaxial stack* with an adjacent base pair, or can form

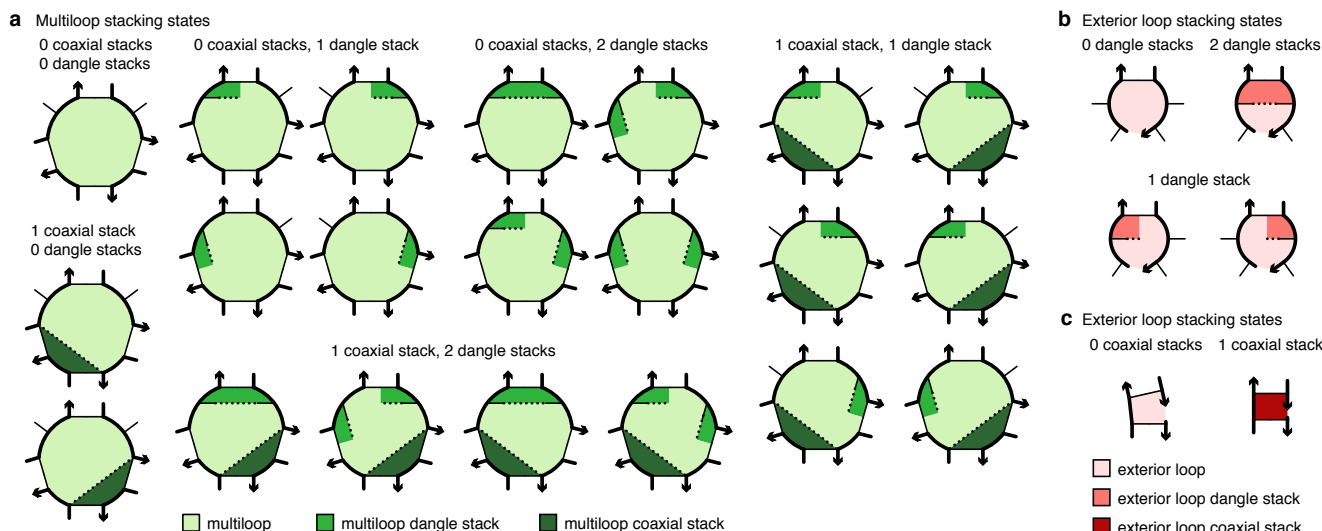


Figure 3. Coaxial and dangle stacking states for multiloops and exterior loops. (a) Stacking subensemble for the multiloop of Figure 2. (b,c) Stacking subensembles for two exterior loops from Figure 2. Adapted with permission from Fornace *et al.*, *ACS Synth Biol*, 9, 2665-2678, 2020. Copyright 2020 American Chemical Society.

a *dangle stack* with at most two adjacent unpaired bases; unpaired bases can either form no stack, or can form a dangle stack with at most one adjacent base pair.

For a given multiloop or exterior loop, the energetic contributions of all possible coaxial and dangle stacking states are enumerated so as to calculate the free energy:⁹

$$\Delta G^{\text{stacking}} = -kT \log \sum_{\omega \in \text{loop}} \prod_{x \in \omega} e^{-\Delta G_x/kT} \quad (2)$$

where ω indexes the possible stacking states within the loop and x indexes the individual stacks (coaxial or dangle) within a stacking state. The free energy of a multiloop or exterior loop is augmented by the corresponding $\Delta G^{\text{stacking}}$ bonus. Hence, a secondary structure s continues to be defined as a set of base pairs, and the stacking states within a given multiloop or exterior loop are treated as a structural subensemble that contributes in a Boltzmann-weighted fashion to the free energy model for the loop. Let $s'' \in s$ denote a stacking state of the paired and unpaired bases in s . We may equivalently define the free energy of secondary structure s in terms of the *stacking state free energies*

$$\overline{\Delta G}(\phi, s'') \quad (3)$$

for all stacking states $s'' \in s$:

$$\overline{\Delta G}(\phi, s) = -kT \log \sum_{s'' \in s} e^{-\overline{\Delta G}(\phi, s'')/kT} \quad (4)$$

Let $\overline{\Gamma}(\phi)$ denote the ensemble of stacking states corresponding to the complex ensemble of secondary structures $\overline{\Gamma}(\phi)$.

NUPACK supports the following coaxial and dangle stacking formulations:

- *All stacking* (default): complex ensemble with coaxial and dangle stacking.
- *Coaxial stacking*: complex ensemble with coaxial stacking.
- *Dangle stacking*: complex ensemble with dangle stacking.
- *No stacking*: complex ensemble without coaxial or dangle stacking.

For backwards compatibility with NUPACK 3, NUPACK 4 also supports historical complex ensembles without coaxial stacking and with approximate dangle stacking.

Symmetry correction. For a secondary structure $s \in \Gamma(\phi)$ with an R -fold rotational symmetry, there is an R -fold reduction in distinguishable conformational space, so the free energy $\overline{\Delta G}(\phi, s)$ must be adjusted⁵ by a symmetry correction:

$$\Delta G(\phi, s) = \overline{\Delta G}(\phi, s) + \Delta G^{\text{sym}}(\phi, s). \quad (5)$$

where

$$\Delta G^{\text{sym}}(\phi, s) = kT \log R(\phi, s). \quad (6)$$

Because the symmetry factor $R(\phi, s)$ is a global property of each secondary structure $s \in \Gamma(\phi)$, it is not suitable for use with dynamic programs that treat multiple subproblems simultaneously without access to global structural information. As a result, dynamic programs operate on ensemble $\overline{\Gamma}(\phi)$ using physical model $\overline{\Delta G}(\phi, s)$ and then the Distinguishability Correction Theorem⁵ enables exact conversion of physical quantities to ensemble $\Gamma(\phi)$ using physical model $\Delta G(\phi, s)$. Interestingly, ensembles $\overline{\Gamma}(\phi)$ and $\Gamma(\phi)$ both have utility when examining the physical properties of a complex as they provide related but different perspectives, akin to complementary thought experiments.⁹

Free energy parameters. NUPACK supports the following temperature-dependent parameter sets for RNA:

- **rna06** based on References 14, 19 and 21 with additional parameters^{13, 17} including coaxial stacking^{14, 22} and dangle stacking^{11, 17, 22} in 1M Na⁺.
- **rna95** based on Reference 11 with additional parameters¹⁷ including coaxial stacking^{14, 22} and dangle stacking^{11, 17, 22} in 1M Na⁺.

and for DNA:

- **dna04** based on References 12 and 18 with additional parameters¹⁷ including coaxial stacking¹⁶ and dangle stacking^{15, 17} in user-specified concentrations of Na⁺, K⁺, NH₄⁺, and Mg⁺⁺.^{12, 16, 18, 20}

DNA/RNA hybrids are not allowed. For backwards compatibility with NUPACK 3, NUPACK 4 also supports historical DNA and RNA parameter sets.

Salts. The default salt conditions for RNA and DNA parameter sets are [Na⁺] = 1M; these are the only salt conditions for RNA. Salt corrections are available for DNA parameters to permit calculations in user-specified sodium, potassium, ammonium, and magnesium ion concentrations.

- *Sodium*: based on refs 12 and 18; the sum of the concentrations of (monovalent) sodium, potassium, and ammonium ions, $[\text{Na}^+] + [\text{K}^+] + [\text{NH}_4^+]$, is specified in units of molar (default: 1.0, range: $[0.05, 1.1]$).
- *Magnesium*: based on refs 16 and 20; the concentration of (divalent) magnesium ions, $[\text{Mg}^{++}]$, is specified in units of molar (default: 0.0, range: $[0.0, 0.2]$).

PHYSICAL QUANTITIES

Consider a multi-tube ensemble Ω where tube $h \in \Omega$ contains a set of strand species Ψ_h^0 interacting to form a set of complex species Ψ_h . Let $j \in \Psi_h$ denote a complex with sequence ϕ_j and complex ensembles $\bar{\Gamma}(\phi_j)$ (treating all strands as distinct) and $\Gamma(\phi_j)$ (treating strands with the same sequence as indistinguishable). NUPACK calculates a number of physical quantities over these ensembles.^{5,9}

Partition function. For complex j , the partition function evaluated over ensemble $\Gamma(\phi_j)$ treating strands with the same sequence as indistinguishable is denoted

$$Q(\phi_j) = \sum_{s \in \Gamma(\phi_j)} e^{-\Delta G(\phi_j, s)/kT}. \quad (7)$$

Complex free energy. For complex j , the corresponding complex free energy is

$$\Delta G(\phi_j) \equiv -kT \log(Q(\phi_j)). \quad (8)$$

Structure free energy. For complex j , the secondary structure free energy treating strands with the same sequence as indistinguishable is denoted

$$\Delta G(\phi_j, s). \quad (9)$$

If the physical model includes coaxial and dangle stacking, the structure free energy will include stacking contributions $\Delta G^{\text{stacking}}$. If the secondary structure s has a rotational symmetry, the structure free energy will include the symmetry correction $\Delta G^{\text{sym}}(\phi_j, s)$.

Equilibrium structure probability. For complex j , the equilibrium structure probability of any secondary structure $s \in \Gamma(\phi_j)$ treating strands with the same sequence as indistinguishable is denoted

$$p(\phi_j, s) = e^{-\Delta G(\phi_j, s)/kT} / Q(\phi_j). \quad (10)$$

Boltzmann-sampled structures. For complex j , a set of J secondary structures Boltzmann-sampled from ensemble $\Gamma(\phi_j)$ treating strands with the same sequence as indistinguishable is denoted

$$\Gamma_{\text{sample}}(\phi_j, J) \in \Gamma(\phi_j). \quad (11)$$

Boltzmann-sampled structures are available only via the NUPACK Python module.

Equilibrium base-pairing probabilities. For complex j , the base-pairing probability matrix $\bar{P}(\phi_j)$ has entries $\bar{P}^{a,b}(\phi_j) \in [0, 1]$ corresponding to the probability

$$\bar{P}^{a,b}(\phi_j) = \sum_{s \in \bar{\Gamma}(\phi_j)} \bar{p}(\phi_j, s) S^{a,b}(s) \quad (12)$$

that base pair $a \cdot b$ forms at equilibrium within ensemble $\bar{\Gamma}(\phi_j)$, treating all strands as distinct. Here, $S(s)$ is the structure matrix and $\bar{p}(s)$ is the equilibrium probability of structure $s \in \bar{\Gamma}(\phi_j)$ treating all strands as distinct. Abusing notation, the entry $\bar{P}^{i,i}(\phi_j) \in [0, 1]$ denotes the equilibrium probability that base i is unpaired over ensemble $\bar{\Gamma}(\phi_j)$. Hence, $\bar{P}(\phi_j)$ is a symmetric matrix with row and column sums of 1. In NUPACK graphics, the diagonal entries denoting unpaired bases are depicted as an extra column to the right of the matrix.

MFE proxy structure. For complex j , the minimum free energy (MFE) stacking state $s_{\text{MFE}}^{\text{II}}(\phi_j) \in \bar{\Gamma}^{\text{II}}(\phi_j)$ treating all strands as distinct is

$$s_{\text{MFE}}^{\text{II}}(\phi_j) = \arg \min_{s^{\text{II}} \in \bar{\Gamma}^{\text{II}}(\phi_j)} \overline{\Delta G}(\phi_j, s^{\text{II}}) \quad (13)$$

The corresponding MFE proxy structure is

$$s_{\text{MFE}'}(\phi_j) \equiv \{s \in \bar{\Gamma}(\phi_j) | s_{\text{MFE}}^{\text{II}}(\phi_j) \in s\}, \quad (14)$$

defined as the secondary structure containing the MFE stacking state within its subensemble. The free energy of the MFE proxy structure is

$$\overline{\Delta G}(\phi_j, s_{\text{MFE}'}(\phi_j)). \quad (15)$$

There may be more than one MFE stacking state, each corresponding to the same or different MFE proxy structures. If there are multiple MFE proxy structures, the NUPACK web app presents only one of them; to see multiple MFE proxy structures, use the NUPACK Python module.

Suboptimal proxy structures. For complex j , the set of suboptimal proxy secondary structures with stacking states within a specified $\Delta G_{\text{gap}} \geq 0$ of the MFE stacking state is denoted

$$\begin{aligned} \bar{\Gamma}_{\text{subopt}}(\phi_j, \Delta G_{\text{gap}}) \\ = \{s \in \bar{\Gamma}(\phi_j) | s^{\text{II}} \in s, \\ \overline{\Delta G}(\phi_j, s^{\text{II}}) \leq \overline{\Delta G}(\phi_j, s_{\text{MFE}}^{\text{II}}(\phi_j)) + \Delta G_{\text{gap}}\}. \end{aligned} \quad (16)$$

Suboptimal proxy structures are available only via the NUPACK Python module.

Complex ensemble defect. For complex j with target structure s_j , the dimensional complex ensemble defect

$$n(\phi_j, s_j) = |\phi_j| - \sum_{\substack{1 \leq a \leq |\phi_j|, \\ 1 \leq b \leq |\phi_j|}} \bar{P}^{a,b}(\phi_j) S^{a,b}(s_j), \quad (17)$$

represents the equilibrium number of incorrectly paired nucleotides over the ensemble $\bar{\Gamma}(\phi_j)$ relative to target structure s_j .^{3,6} Here, $\bar{P}(\phi_j)$ is the equilibrium base-pairing probability matrix and $S(s_j)$ is the target structure matrix for s_j . The *normalized complex ensemble defect* is then

$$\mathcal{N}_j \equiv n(\phi_j, s_j) / |\phi_j| \in [0, 1], \quad (18)$$

representing the equilibrium fraction of incorrectly paired nucleotides evaluated over the ensemble of complex j relative to target structure s_j .

Complex ensemble size. For complex j , the number of secondary structures in the complex ensemble, treating all strands as distinct, is denoted

$$|\bar{\Gamma}(\phi_j)|. \quad (19)$$

The corresponding number of stacking states is denoted

$$|\bar{\Gamma}^{\text{II}}(\phi_j)|. \quad (20)$$

Equilibrium complex concentrations. For the set of complexes Ψ_h in test tube h , the set of equilibrium complex concentrations is denoted

$$x_{\Psi_h} \equiv x_j \quad \forall j \in \Psi_h. \quad (21)$$

These concentrations are the unique solution to the strictly convex optimization problem⁵

$$\min_{x_{\Psi_h}} \sum_{j \in \Psi_h} x_j (\log x_j - \log Q_j - 1) \quad (22)$$

$$\text{subject to} \quad \sum_{j \in \Psi_h} A_{i,j} x_j = x_i^0 \quad \forall i \in \Psi_h^0, \quad (23)$$

expressed in terms of the previously calculated set of partition functions Q_{Ψ_h} . Here, the constraints impose conservation of mass: A is the stoichiometry matrix such that $A_{i,j}$ is the number of strands of type i in complex j , and x_i^0 is the total concentration of strand i present in the test tube. Based on dimensional analysis,⁵ the convex optimization problem is formulated in terms of mole fractions, but for convenience, NUPACK accepts molar strand concentrations $[i]^0 = x_i^0[\text{H}_2\text{O}]$ as inputs and returns molar complex concentrations $[j] = x_j[\text{H}_2\text{O}]$ as outputs, where $[\text{H}_2\text{O}]$ is the molarity of water. Hence, the user specifies the set of molar strand concentrations $[i]^0 \forall i \in \Psi_h^0$ and NUPACK calculates the set of molar complex concentrations $[j] \forall j \in \Psi_h$.

Test tube fraction of bases unpaired. For a test tube $h \in \Omega$ containing the set of complexes Ψ_h , the fraction of bases unpaired

$$f_h^{\text{unpaired}} \in [0, 1] \quad (24)$$

denotes the fraction of bases that are unpaired in tube h at equilibrium, which is calculated based on the set of equilibrium concentrations x_{Ψ_h} and the set of base-pairing probability matrices \bar{P}_{Ψ_h} .

Test tube ensemble pair fractions. For a test tube $h \in \Omega$ containing the set of complexes Ψ_h , the ensemble pair fraction

$$f_h^{\text{A}}(a_A \cdot b_B) \quad (25)$$

denotes the fraction of A strands that form base pair $a_A \cdot b_B$ in tube h . Correspondingly,

$$f_h^{\text{B}}(a_A \cdot b_B) \quad (26)$$

denotes the fraction of B strands that form base pair $a_A \cdot b_B$ in tube h . These base-pairing observables depend on the set of equilibrium concentrations x_{Ψ_h} and the set of base-pairing probability matrices \bar{P}_{Ψ_h} . The number of distinct bases in the test tube is:

$$N_{\text{distinct}} \equiv \sum_{i=1}^{|\Psi_h^0|} |\phi_i| \quad (27)$$

representing the total number of bases in all $|\Psi_h^0|$ strand species. Numbering the distinct bases from 1 to N_{distinct} , the ensemble pair fractions are then stored as an (asymmetric) $N_{\text{distinct}} \times N_{\text{distinct}}$ matrix with the fraction of each nucleotide that is unpaired stored on the diagonal. Hence, the matrix of test tube ensemble pair fractions is asymmetric with row and column sums of 1. In NUPACK graphics, the diagonal entries denoting unpaired bases are depicted as an extra column to the right of the matrix.

Test tube ensemble defect. Consider test tube $h \in \Omega$ containing a set of desired *on-target complexes*, Ψ_h^{on} , and a set of undesired *off-target complexes*, Ψ_h^{off} . The set of complexes in the test tube is then:

$$\Psi_h = \Psi_h^{\text{on}} \cup \Psi_h^{\text{off}}. \quad (28)$$

Let each on-target complex, $j \in \Psi_h^{\text{on}}$, have a user-specified target secondary structure, s_j , and a user-specified target concentration, $y_{h,j}$. Let each off-target complex, $j \in \Psi_h^{\text{off}}$, have a vanishing target concentration ($y_{h,j} = 0$) and no target structure ($s_j = \emptyset$). The dimensional test tube ensemble defect,

$$\begin{aligned} C(\phi_{\Psi_h}, s_{\Psi_h}, y_{h,\Psi_h}) &= \sum_{j \in \Psi_h^{\text{on}}} \left[n(\phi_j, s_j) \min(x_{h,j}, y_{h,j}) \right. \\ &\quad \left. + |\phi_j| \max(y_{h,j} - x_{h,j}, 0) \right] \quad (29) \end{aligned}$$

represents the equilibrium concentration of incorrectly paired nucleotides over the ensemble of test tube h .⁷ Here, $x_{h,j}$ is the equilibrium concentration of complex j in tube h . For each on-target complex, $j \in \Psi_h^{\text{on}}$, the first term in the sum represents the *structural defect*, quantifying the concentration of nucleotides that are in an incorrect base-pairing state within the ensemble of complex j , and the second term in the sum represents the *concentration defect*, quantifying the concentration of nucleotides that are in an incorrect base-pairing state because there is a deficiency in the concentration of complex j . For each off-target complex, $j \in \Psi_h^{\text{off}}$, the structural and concentration defects are identically zero, since $y_{h,j} = 0$. This does not mean that the defects associated with off-targets are ignored. By conservation of mass, non-zero off-target concentrations imply deficiencies in on-target concentrations, and these concentration defects are quantified by the equation above.⁷ The *normalized test tube ensemble defect* is then denoted

$$\mathcal{M}_h \equiv C_h / y_h^{\text{nt}} \in [0, 1] \quad (30)$$

representing the equilibrium fraction of incorrectly paired nucleotides in tube h . Here,

$$y_h^{\text{nt}} \equiv \sum_{j \in \Psi_h^{\text{on}}} |\phi_j| y_{h,j} \quad (31)$$

is the total concentration of nucleotides in tube h . As \mathcal{M}_h approaches zero, each on-target complex, $j \in \Psi_h^{\text{on}}$, approaches its target concentration, $y_{h,j}$, and is dominated by its target structure, s_j , and each off-target complex, $j \in \Psi_h^{\text{off}}$, forms with vanishing target concentration.

For the set of test tubes Ω , the *multi-tube ensemble defect*

$$\mathcal{M} \equiv \frac{1}{|\Omega|} \sum_{h \in \Omega} \mathcal{M}_h \in (0, 1) \quad (32)$$

represents the average equilibrium fraction of incorrectly paired nucleotides over the test tubes $h \in \Omega$.

DESIGN FORMULATION

NUPACK provides a framework for engineering reaction pathways for dynamic hybridization cascades (e.g., shape and sequence transduction using small conditional RNAs^{23,24}) or for engineering large-scale structures including pseudoknots (e.g., RNA origamis²⁵). In either case, sequence design is performed over a multi-tube ensemble (see Figure 4 for a cautionary tale emphasizing the advantages of test tube design over complex design).^{7,8}

For reaction pathway engineering, sequence design is formulated as a multistate optimization problem using a set of target test tubes to represent elementary steps of the reaction pathway, as well as to model crosstalk between components. Note that *kinetic design* of a test tube ensemble is achieved by performing *equilibrium optimization* of a multi-tube ensemble: each target test tube isolates different subsets of components in local equilibrium, enabling optimization of kinetically significant states that would appear insignificant if all components were allowed to interact in a single ensemble. For large-scale structural engineering including the possibility of pseudoknots, each target test tube contains only complex ensembles comprising unpseudoknotted structures, but by imposing sequence constraints between tubes, it is possible to collectively impose pseudoknotted design requirements.

In a multi-tube design ensemble, each target test tube contains a set of desired *on-target complexes*, each with a target

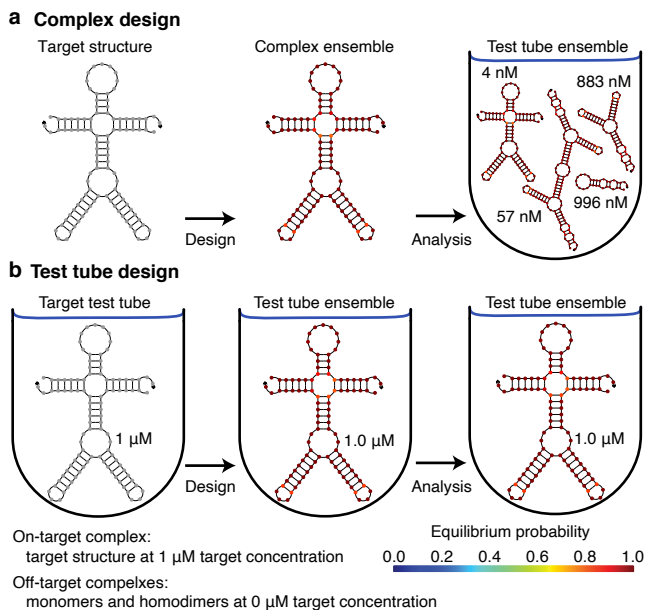


Figure 4. A cautionary tale: the advantages of test tube design over complex design. (a) Complex design. Sequence design formulated in the context of a complex (left) ensures that at equilibrium the target structure dominates the structural ensemble of the complex (center). Unfortunately, subsequent test tube analysis reveals that the desired on-target complex occurs at negligible concentration relative to other undesired off-target complexes (right). With complex design, neither the concentration of the desired on-target complex, nor the concentrations of undesired off-target complexes are considered. As a result, sequences that are successfully optimized to predominantly adopt a target secondary structure in the context of an on-target complex, may nonetheless fail to ensure that this complex forms at appreciable concentration when the strands are introduced into a test tube. (b) Test tube design. Sequence design formulated in the context of a test tube (left) ensures that at equilibrium the desired on-target complex is dominated by its target structure and forms at approximately its target concentration, and that undesired off-target complexes form at negligible concentrations (center). Subsequent test tube analysis (right) provides no new information and no unpleasant surprises since the design and analysis ensembles are identical. Adapted with permission from Wolfe and Pierce *ACS Synth Biol*, 4, 1086-1100, 2015. Copyright 2014 American Chemical Society.

secondary structure and target concentration, and a set of undesired *off-target complexes*, each with vanishing target concentration. Optimization of the *multi-tube ensemble defect*, representing the average equilibrium fraction of incorrectly paired nucleotides evaluated over the design ensemble, implements both a positive design paradigm, explicitly designing for on-pathway elementary steps, and a negative design paradigm, explicitly designing against off-pathway crosstalk. *Defect weights* can be specified to prioritize or de-prioritize design quality for different portions of the design ensemble. Sequence design is performed subject to user-specified *hard constraints* that prohibit sequences violating the constraints and *soft constraints* that penalize (but do not prohibit) sub-optimal sequences.

Reaction pathways. Consider a set of nucleic acid molecules intended to execute a prescribed hybridization cascade.⁸ For example, the reaction pathway of Figure 5 describes small conditional RNAs (scRNAs) that upon binding to input X, perform shape and sequence transduction to form a Dicer substrate targeting an independent output Y for silencing.²³ A *reaction pathway* specifies the *elementary steps* (each a

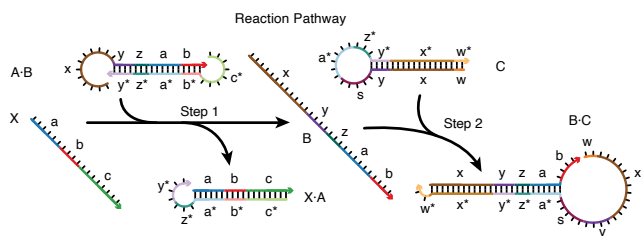


Figure 5. Reaction pathway schematic. Conditional Dicer substrate formation via shape and sequence transduction with small conditional RNAs (scRNAs).²³ scRNA A-B detects input X (comprising sequence “a-b-c”), leading to production of Dicer substrate B-C (targeting independent sequence “w-x-y-z”). Step 1: X displaces A from B via toehold-mediated 3-way branch migration and spontaneous dissociation. Step 2: B assembles with C via loop/toehold nucleation and 3-way branch migration to form Dicer substrate B-C. See⁸ for additional reaction pathway case studies. Adapted with permission from Wolfe *et al.*, *J Am Chem Soc*, 139, 3134-3144, 2017. Copyright 2017 American Chemical Society.

self-assembly or disassembly operation in which complexes form or break) by which the molecules are intended to interact, the desired secondary structure for each on-pathway complex, and the complementarity relationships between sequence domains in the molecules. In the reaction pathway of Figure 5, there are two elementary steps (Step 1: X + A-B → X-A + B, Step 2: B + C → B-C) involving six on-pathway complexes (X, A-B, X-A, B, C, B-C) and numerous sequence domains (“a*” reverse complementary to “a”, “b*” reverse complementary to “b”, and so on).

In addition to specifying a set of desired on-pathway elementary steps, each reaction pathway also implicitly specifies a much larger set of off-pathway interactions, corresponding to undesired *crosstalk* between components within the pathway or with components from other unrelated reaction pathways. To perform sequence design for reaction pathway engineering, we formulate a multistate optimization problem to explicitly design for on-pathway elementary steps (a positive design paradigm) and against off-pathway crosstalk (a negative design paradigm).⁸

Multi-tube design ensemble. A multi-tube design problem is specified as a set of *target test tubes*, Ω .⁸ Each tube, $h \in \Omega$, contains a set of desired *on-target complexes*, Ψ_h^{on} , and a set of undesired *off-target complexes*, Ψ_h^{off} . For each on-target complex, $j \in \Psi_h^{\text{on}}$, the user specifies a target secondary structure, s_j , and a target concentration, $y_{h,j}$. For each off-target complex, $j \in \Psi_h^{\text{off}}$, the target concentration is vanishing ($y_{h,j} = 0$) and there is no target structure ($s_j = \emptyset$). Note that complex j may have a different target concentration, $y_{h,j}$, in each tube h (e.g., may be an on-target in one tube and an off-target in another tube). By contrast, complex j has the same target structure, s_j , in all tubes where it appears as an on-target (the target structure is ignored in any tubes where complex j appears as off-target).

The set of complexes in tube h is then $\Psi_h \equiv \Psi_h^{\text{on}} \cup \Psi_h^{\text{off}}$ and the set of all complexes in multistate test tube ensemble Ω is $\Psi \equiv \cup_{h \in \Omega} \Psi_h$. Let

$$\phi_\Psi \equiv \phi_j \quad \forall j \in \Psi \quad (33)$$

denote the set of sequences for the complexes in Ψ .

Consider specification of the multi-tube ensemble, Ω , for the design of N orthogonal systems for a reaction pathway of M elementary steps. One *elementary step tube* is specified for each step $m = 0, \dots, M$ for each system $n = 1, \dots, N$ (treating formation of the initial reactants as a precursor “Step 0”). Additionally, a single *global crosstalk tube* is specified to minimize off-pathway interactions between the reactive species

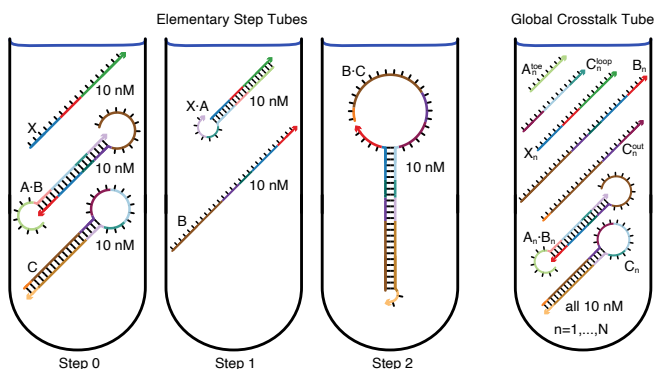


Figure 6. Target test tubes. Left: Elementary step tubes. Step 0 tube: target X and scRNAs A-B and C. Step 1 tube: X-A and B. Step 2 tube: Dicer substrate B-C. Each target test tube contains the depicted on-target complexes corresponding to the on-pathway products for a given step (each with the depicted target secondary structure and a target concentration of 10 nM) as well as off-target complexes (not depicted) corresponding to on-pathway reactants and off-pathway crosstalk for a given step. To design N orthogonal systems, there are three elementary step tubes for each system $n = 1, \dots, N$. Right: Global crosstalk tube. Contains the depicted on-target complexes corresponding to reactive species generated during Steps 0, 1, 2 as well as off-target complexes (not depicted) corresponding to off-pathway interactions between these reactive species. To design N orthogonal systems, the global crosstalk tube contains a set of on-targets and off-targets for each system $n = 1, \dots, N$. Adapted with permission from Wolfe *et al.*, *J Am Chem Soc*, 139, 3134-3144, 2017. Copyright 2017 American Chemical Society.

generated during all elementary steps of all systems. The total number of target test tubes is then $|\Omega| = N \times (M + 1) + 1$.

Target test tubes. Figure 6 depicts target test tubes for the reaction pathway of Figure 5. There are three *elementary step tubes*, each containing on-target complexes corresponding to the products of the corresponding step: the Step 0 tube contains on-targets X, A-B, and C; the Step 1 tube contains on-targets X-A and B; the Step 2 tube contains on-target B-C. Each elementary step tube contains a set of on-target complexes (each with a target secondary structure and target concentration), corresponding to the on-pathway hybridization products for a given step, and a set of undesired off-target complexes (each with vanishing target concentration), corresponding to on-pathway reactants and off-pathway hybridization crosstalk for a given step. Hence, these elementary step tubes design for full conversion of cognate reactants into cognate products and against local crosstalk between these same reactants.

To simultaneously design N orthogonal systems, three elementary step tubes of the type shown in Figure 6 (left) are specified for each system. Furthermore, to design against off-pathway interactions within and between systems, a single *global crosstalk tube* is specified (right). In the global crosstalk tube, the on-target complexes correspond to all reactive species generated during all elementary steps ($m = 0, 1, 2$) for all systems ($n = 1, \dots, N$); the off-target complexes correspond to non-cognate interactions between these reactive species. Crucially, the global crosstalk tube ensemble omits the cognate products that the reactive species are intended to form (they appear as neither on-targets nor off-targets). Hence, all reactive species in the global crosstalk tube are forced to either perform no reaction (remaining as desired on-targets) or undergo a crosstalk reaction (forming undesired off-targets), providing the basis for minimization of global crosstalk during sequence optimization. To design 8 orthog-

onal systems for this reaction pathway, the total number of target test tubes is then $|\Omega| = 8 \times 3 + 1 = 25$. See Section S2.2 of Reference 8 for a general description of how to specify target test tubes for a given reaction pathway and a number of illustrative case studies.

Note that each target test tube isolates a different subset of the system components in local equilibrium, enabling optimization of kinetically significant states that would appear insignificant if all components were allowed to interact in a single ensemble. For example, the Step 1 tube in Figure 6 simultaneously optimizes for high-yield production of unstructured intermediate B and against appreciable formation of off-target dimer B-B, promoting rapid nucleation of the unstructured toehold in B with the loop of hairpin C during the next step of the reaction pathway.

Note also that for a tube containing a given set of system components, the cognate products of their interactions can be excluded from the ensemble (appearing as neither on-targets nor off-targets), enabling optimization for high-yield well-structured reactants and against crosstalk. For example, in Figure 6, the Step 0 tube excludes the cognate products of Step 1 (X-A and B) from the ensemble in order to optimize formation of initial reactants X, A-B, and C and discourage competing crosstalk interactions (e.g., X-X, A-A, X-C).

Design objective function. The design objective function is the multi-tube ensemble defect,⁸

$$\mathcal{M} \in [0, 1], \quad (34)$$

representing the average equilibrium fraction of incorrectly paired nucleotides over the multi-tube ensemble, Ω . Test tube ensemble defect optimization implements a positive design paradigm (stabilize on-targets) and a negative design paradigm (destabilize off-targets) at two levels: a) designing for the on-target structure and against all off-target structures within the structural ensemble of each on-target complex,^{3,6} and b) designing for the target concentration of each on-target complex and against the formation of all off-target complexes within the ensemble of the test tube.^{7,8} Both paradigms are crucial at both levels in order to achieve high-quality test tube designs with a low test tube ensemble defect.^{7,8}

Defect weights. To prioritize or de-prioritize design quality for a portion of the design ensemble, the defect-weighted objective function, \mathcal{M}_W , incorporates user-specified defect weights, W , for the multi-tube ensemble or for any domain, strand, complex, or tube. With the default value of unity for all weights, \mathcal{M}_W is simply the multi-tube ensemble defect, \mathcal{M} . With custom defect weights in the range $[0, \infty)$, the physical meaning of the objective function is distorted in the service of adjusting design priorities. Increasing the weight for a tube, complex, strand or domain will lead to a corresponding increase in the allocation of effort to designing this entity, typically leading to a corresponding reduction in the defect contribution of the entity. Likewise, decreasing the weight for a domain, strand, complex, or tube will lead to a corresponding decrease in the allocation of effort to designing this entity, typically leading to a corresponding increase in the defect contribution of the entity.

Hard constraints. Sequence design can be performed subject to *hard constraints* that prohibit sequences violating the constraints. NUPACK supports the following types of hard constraints:^{8,9}

- *Assignment constraints:* Constrain consecutive nucleotides to have a specified sequence (specified 5' to 3' using degenerate nucleotide codes; see Table 2); specify an assignment constraint by defining a *domain*.

- *Match constraints*: Force a concatenation of one list of domains and an equal-length concatenation of another list of domains to have identical sequences.
- *Complementarity constraints*: Force a concatenation of one list of domains to be the reverse Watson-Crick complement of an equal-length concatenation of another list of domains (optionally allow wobble mutations for RNA designs [yielding G-U base pairs]). Note that nucleotides that are base-paired in the target structure of an on-target complex are automatically assigned a complementarity constraint.
- *Diversity constraints*: Force every word of a specified length to contain a specified degree of sequence diversity, either globally or for a concatenated list of domains (e.g., require every subsequence of length 4 to have at least 2 nucleotide types).
- *Similarity constraints*: Force a concatenation of a list of domains to match an equal-length reference sequence to within a specified fractional range (e.g., require 45%-55% GC content by imposing a similarity constraint of 45%-55% to a sequence that is poly-S).
- *Window constraints*: Force a concatenation of a list of domains to be a subsequence of a source sequence (e.g., the source sequence is an mRNA), or more generally, a subsequence of one of multiple source sequences.
- *Library constraints*: Force a concatenation of a list of domains to have sequences drawn from a concatenated list of libraries. Each library contains a set of alternative sequences of equal length (e.g., a library of toehold sequences or a library of codons).
- *Pattern constraints*: Prevent a list of patterns from appearing globally or in a concatenation of a list of domains (e.g., prevent GGGG, which is prone to forming G-quadruplexes that are not accounted for in the empirical physical model).

Let \mathcal{R} denote the user-specified set of hard constraints for a design problem.

Soft constraints. As an alternative to hard constraints that prohibit constraint violations, *soft constraints* define auxiliary objective functions that penalize suboptimal sequences during the design process:²⁸

$$w_k f_k(\phi_\Psi). \quad (35)$$

Here, $f_k(\phi_\Psi) \in [0, 1]$ is the penalty function for soft constraint k and $w_k \in [0, \infty)$ (default: 1) is the corresponding user-specified weight. Soft constraints can reduce design cost relative to the corresponding hard constraint by making it easier for the optimization process to identify candidate sequence mutations. Soft constraints can also increase flexibility by enabling specification of new design goals (e.g., designing two toeholds to have comparable binding strength) for which there is no hard constraint analog. NUPACK supports the following types of soft constraints:²⁸

- *Similarity constraints*: Penalize a concatenation of a list of domains if it does not match an equal-length reference sequence to within a specified fractional range (e.g., prioritize 45%-55% GC content by imposing a similarity constraint of 45%-55% to a sequence that is poly-S).
- *Pattern constraints*: Penalize a list of patterns if they appear globally or in a concatenation of a list of domains (e.g., penalize GGGG, which is prone to forming G-quadruplexes that are not accounted for in the empirical physical model).
- *Symmetry constraints*: Penalize a subsequence of a specified word length:²⁹ 1) if the word appears in more than

one location in the design, 2) if the reverse complement of the word appears elsewhere in a location that is not intended to form a duplex with the word, or 3) if the word is self-complementary.

- *Energy match constraints*: Penalize a set of duplexes (e.g., toeholds and toehold complements) if their structure free energies deviate from each other, or alternatively from a specified reference free energy.

Let \mathcal{S} denote the user-specified set of soft constraints for a design problem.

Constrained multi-tube design problem. To design a set of sequences, Φ_Ψ , for a multi-tube ensemble, Ω , subject to user-specified hard constraints \mathcal{R} and soft constraints \mathcal{S} , the constrained multi-tube design problem is:

$$\min_{\phi_\Psi} \left[\mathcal{M}_\mathcal{W} + \sum_{k \in \mathcal{S}} w_k f_k(\phi_\Psi) \right] \quad \text{subject to } \mathcal{R}, \quad (36)$$

where $\mathcal{M}_\mathcal{W}$ is the multi-tube ensemble defect including user-specified defect weights \mathcal{W} . The sequence design algorithm seeks to iteratively reduce the *augmented objective function* (weighted ensemble defect plus weighted soft constraints) below the *stop condition*

$$\left[\mathcal{M}_\mathcal{W} + \sum_{k \in \mathcal{S}} w_k f_k(\phi_\Psi) \right] \leq f_{\text{stop}} \quad (37)$$

for user-specified $f_{\text{stop}} \in (0, 1)$ while satisfying the hard constraints in \mathcal{R} .

ANALYSIS PAGE

The Analysis page of the NUPACK web app enables users to analyze the equilibrium concentration and base-pairing properties of a multi-tube ensemble containing one or more test tubes. Each test tube ensemble contains a user-specified set of strand species, each introduced at a user-specified concentration.

Input. The Analysis Input page allows the user to specify the physical model and components for the multi-tube ensemble. For the multi-tube ensemble, specify the following:

- *Material*: Select RNA or DNA.
- *Temperature*: Specify the temperature in Celsius (or select “Melt” and specify a minimum temperature, increment, and maximum temperature to simulate the multi-tube ensemble for a range of temperatures).
- *Model options*: Optionally specify details of the physical model:
 - *Parameters*: Select from available free energy parameter sets.
 - *Ensemble*: Specify the coaxial and dangle stacking formulation.
 - *Salts*: Specify salt concentrations (Na^+ and Mg^{++}).

For each test tube within the multi-tube ensemble, specify the following:

- *Tube name*: Specify the name of the test tube.
- *Strands*: Specify the name, sequence, and concentration of each strand species.
- *Complexes*: Specify the complex species in the test tube in any of three ways:
 - *Max complex size*: Automatically generate all complexes up to a specified maximum number of strands (default: 1).
 - *Include complex*: explicitly specify complexes to include in the test tube ensemble (that would otherwise not be included based on the specified maximum complex size).

- *Exclude complex*: explicitly specify complexes to exclude from the test tube ensemble (that would otherwise be included based on the specified maximum complex size).

Computation. For each complex in the multi-tube ensemble, the partition function, equilibrium base-pairing probabilities, and minimum free energy (MFE) proxy structure are calculated using a unified dynamic programming framework.^{5,9} If the same strand species are present in more than one tube of a multi-tube ensemble, the algorithms achieve significant cost savings relative to analyzing each tube in the ensemble separately.⁹ For each test tube ensemble, the equilibrium complex concentrations are the solutions to a strictly convex optimization problem (formulated in terms of calculated partition functions and user-specified strand concentrations), which we solve efficiently in the dual form.⁵ The equilibrium complex concentrations and base-pairing probabilities are then used to calculate the test tube fraction of bases unpaired and test tube ensemble pair fractions.⁵ In order to analyze the tube properties for a different set of strand concentrations, it is not necessary to rerun the dynamic programs to calculate partition functions, equilibrium base-pairing probabilities, and MFE proxy structures; only the convex optimization problem must be solved again, and this can be rapidly done from within the Results page.

Results. The Analysis Results page summarizes the equilibrium concentration and base-pairing properties of each test tube in the multi-tube ensemble; use the dropdown to examine the results for a given tube.

- *Temperature slider*: For calculations that specified a temperature range, use the temperature slider to examine results over the range of simulated temperatures.
- *Melt profile*: For calculations that specified a temperature range, the melt profile depicts the equilibrium fraction of bases unpaired in the test tube as a function of temperature.
- *Equilibrium complex concentrations*: The bar graph depicts the equilibrium concentration of each complex that forms with appreciable concentration in the test tube (adjust the display filters to alter which complex concentrations are shown). Clicking any bar to display equilibrium base-pairing information for the corresponding complex:
 - *MFE structure*: Depicts the MFE proxy structure for the complex. In the default view, each base is shaded with the probability that it adopts the depicted base-pairing state at equilibrium, revealing which portions of the structure usefully summarize equilibrium structural features of the complex ensemble.
 - *Pair probabilities*: Depicts equilibrium base-pairing probabilities for the complex. By definition, these data are independent of concentration and of all other complexes in solution. The area and color of each dot scale with the equilibrium probability of the corresponding base pair. With this convention, the matrix is symmetric, as denoted by a diagonal line. In the column at right, the area and color of each dot scale with the equilibrium probability that the corresponding base is unpaired within the complex ensemble. Optional black circles depict each base pair or unpaired base in the MFE proxy structure.
- *Test tube ensemble pair fractions*: Depicts equilibrium base-pairing information for the test tube ensemble, taking into account the equilibrium concentration and base-pairing properties of each complex. The area and color of the dot at row i and column j scale with the equilib-

rium fraction of base i that is paired to base j in solution. With this convention, the matrix can be asymmetric. In the column at right, the area and color of the dot in row i scales with the equilibrium fraction of base i that is unpaired within the test tube ensemble.

Information can be exported to the Design or Utilities pages:

- *To Design*: For a given complex, export the MFE proxy structure to the Design page to redesign the sequence.
- *To Utilities*: For a given complex, export the MFE proxy structure and sequence information to the Utilities page to annotate publication quality graphics, or to do quick analysis or design calculations in the context of the complex ensemble.

For individual plots, download graphics for editing in vector graphics programs or download data for local plotting. Alternatively, all job data and plots can be downloaded as a single compressed file.

DESIGN PAGE

The Design page of the NUPACK web app allows users to perform sequence design over a multi-tube ensemble comprising one or more target test tubes. See the Design Formulation section for details on how to formulate a multi-tube design problem.

Input. The Design Input page allows the user to specify the physical model and components for the multi-tube ensemble. For the multi-tube ensemble, specify the following:

- *Material*: Select RNA or DNA.
- *Temperature*: Specify the temperature in Celsius.
- *Trials*: Specify the number of independent design trials.
- *Model options*: Optionally specify details of the physical model:
 - *Parameters*: Select from available free energy parameter sets.
 - *Ensemble*: Specify the coaxial and dangle stacking formulation.
 - *Salts*: Specify salt concentrations (Na^+ and Mg^{++}).
- *Algorithm settings*: Optionally specify algorithm settings:
 - *Stop condition*: Specify a stop condition in the range (0,1). The design algorithm will attempt to reduce the augmented objective function (weighted ensemble defect plus weighted soft constraints) below the stop condition while satisfying hard constraints.
 - *Max design time*: Specify the maximum design time.
 - *Random seed*: Specify a non-zero integer for a reproducible design trial (default: 0; corresponding to a random trial).
 - *Wobble mutations*: For RNA designs, globally prohibit (default) or allow wobble mutations (yielding G-U base pairs). Note that for RNA designs, wobble mutations can also be allowed locally when specifying complementarity hard constraints.

Specify all components that appear in one or more target test tubes within the multi-tube ensemble:

- *Domains*: Specify sequence domains. A domain is a set of consecutive nucleotides that appear as a subsequence of one or more strands in the design, specified as a name and a sequence (specified 5' to 3' using degenerate nucleotide codes; see Table 2). Note that specification of a domain using degenerate nucleotide codes represents an implicit hard sequence constraint.
- *Strands*: Specify target strands. Each target strand is a single RNA or DNA molecule specified as a name and a sequence (specified 5' to 3' in terms of previously specified domains).

- *Target complexes*: Specify target complexes. Each target complex is an on-target and/or off-target complex specified as a name and an ordered list of strands (i.e., an ordering of strands around a circle in a polymer graph) and a complex name. If the complex is to be used as an on-target complex in at least one target test tube, it is specified with an on-target secondary structure (specified in dot-parens-plus, RLE dot-parens-plus, or DU+ notation); the target structure will be ignored in target test tubes where a complex appears as an off-target complex.

For each target test tube in the multi-tube ensemble, specify the following:

- *Target tube name*: Specify the name of the target test tube.
- *On-target complexes*: Specify a set of on-target complexes (from the previously specified set of target complexes that include a target secondary structure), each with a target concentration.
- *Off-target complexes*: Specify off-target complexes in any of three ways:
 - *Max complex size*: Automatically generate the set of all off-target complexes up to a specified maximum number of strands (default: 1).
 - *Include complex*: Explicitly specify off-target complexes to include in the test tube ensemble (that would otherwise not be included based on the specified “Max complex size”).
 - *Exclude complex*: Explicitly specify off-target complexes to exclude from the test tube ensemble (that would otherwise be included based the specified “Max complex size”).

Note that any complex included as an on-target complex will not be included as an off-target complex. Note also that if an off-target is specified using a target complex for which a target structure has been specified, the target structure is ignored (by definition, there is no target structure for an off-target complex). Note further that used together, “Max complex size” and “Exclude complex” provide a powerful combination for specifying target test tubes. With “Max Complex Size” it is possible to specify a large set of off-target complexes formed from a set of system components. With “Exclude complex” it is further possible to remove from this large set all of the cognate products that should form between these system components (so they appear as neither on-targets nor off-targets in the tube ensemble). For example, with this approach, the reactive species in a global crosstalk tube can be forced to either perform no reaction (remaining as desired on-targets) or to undergo a crosstalk reaction (forming undesired off-targets), enabling minimization of global crosstalk during sequence optimization.

Optionally specify hard constraints, soft constraints, and/or defect weights for the multi-tube ensemble:

- *Hard constraints*: Specify hard constraints that prohibit sequences that violate the constraints, including match constraints, complementarity constraints, diversity constraints, similarity constraints, window constraints, library constraints, and pattern constraints.
- *Soft constraints*: Specify soft constraints that penalize (but do not prohibit) suboptimal sequences, including similarity constraints, pattern constraints, symmetry constraints, and energy match constraints.
- *Defect weights*: Specify defect weights to prioritize or de-prioritize design quality for any combination of domain, strand, complex, or tube. Note that a defect weight can be specified as either an absolute weight or as a multiplier of existing weights.

See the Design Formulation section for details.

Computation. The sequence design algorithm seeks to iteratively reduce the augmented objective function (weighted multi-tube ensemble defect plus weighted soft constraints) below a stop condition while satisfying the specified hard constraints. During sequence optimization, candidate mutations to a random initial sequence are efficiently evaluated over the multi-tube ensemble by estimating the multi-tube ensemble defect using test tube ensemble focusing, hierarchical ensemble decomposition, and conditional physical quantities calculated within subensembles.⁶⁻⁸ The progress page displays, for each independent design trial, the augmented objective function as a function of design time.

Results. The following two plots summarize the design results for each independent design trial:

- *Augmented objective function*: This plot displays, for each independent design trial, the augmented objective function comprising:
 - The *weighted objective function*, incorporating any defect weights specified by the user. With the default value of unity for all weights, this reduces to the multi-tube ensemble defect, representing the average equilibrium fraction of incorrectly paired nucleotides over the multi-tube ensemble.
 - The *weighted soft constraint* contribution for each soft constraint type specified by the user.
- *Multi-tube ensemble defect*: This plot displays, for each independent design trial, the multi-tube ensemble defect, representing the average equilibrium percentage of incorrectly paired nucleotides over the multi-tube ensemble. For each design trial, the defect contributions within the multi-tube ensemble come in two varieties:
 - The *structural defect* component quantifies the fraction of nucleotides that are in the incorrect base-pairing state within the correct complex.
 - The *concentration defect* component quantifies the fraction of nucleotides that are in an incorrect base-pairing state because they are not in the correct complex.

Click on the bar for any design trial to explore details for that design trial:

- *Tube defects*: This plot displays, for each target test tube, the *test tube ensemble defect*, representing the equilibrium concentration of incorrectly paired nucleotides over the ensemble of the test tube. For each target test tube, the defect contributions come in two varieties:
 - The *structural defect* component represents the equilibrium concentration of nucleotides that are in the incorrect base-pairing state within the correct complex.
 - The *concentration defect* component represents the equilibrium concentration of nucleotides that are in an incorrect base-pairing state because they are not in the correct complex.

- *Sequences*: Sequence design results are displayed for each sequence domain and each strand in the design ensemble.

Click on the bar for any tube to explore details for that tube:

- *On-target complex contribution to tube defect*: This plot displays, for each on-target complex, the contribution to the test tube ensemble defect, representing the equilibrium concentration of incorrectly paired nucleotides over the ensemble of the test tube. For each on-target complex, the defect contributions come in two varieties:
 - The *structural defect* component represents the equilibrium concentration of nucleotides that are in the incorrect base-pairing state within the ensemble of the complex.
 - The *concentration defect* component represents the equilibrium concentration of nucleotides that are in

an incorrect base-pairing state because there is a deficiency in the concentration of the complex.

- *On-target complex defect*: This plot displays, for each on-target complex in the test tube, the *complex ensemble defect*, representing the equilibrium number of incorrectly paired nucleotides over the ensemble of the complex.
- *On-target complex concentration*: This plot displays, for each on-target complex in the test tube, the *equilibrium complex concentration* and the target concentration.
- *Off-target complex concentration*: This plot displays the equilibrium complex concentration for each off-target complex that forms appreciably in the test tube.

Click on the bar for any on-target complex to explore details for that complex:

- *Target structure*: Depicts the target secondary structure for the on-target complex. By default, each base is shaded with the probability that it adopts the depicted base-pairing state at equilibrium within the complex ensemble. Optionally, each base is shaded according to its identity.
- *Pair probabilities*: Depicts equilibrium base-pairing probabilities for the on-target complex. By definition, these data are independent of concentration and of all other complexes in solution. The area and color of each dot scale with the equilibrium probability of the corresponding base pair. With this convention, the matrix is symmetric, as denoted by a diagonal line. In the column at right, the area and color of each dot scale with the equilibrium probability that the corresponding base is unpaired within the complex ensemble. Optional black circles depict each base pair or unpaired base in the target structure.

Information can be exported to the Analysis or Utilities pages:

- *To Analysis*: For the multi-tube ensemble, a given target test tube, or a given on-target complex, export the designed sequences to the Analysis page for further equilibrium analysis.
- *To Utilities*: For a given on-target complex, export the target structure and designed sequences to the Utilities page to annotate publication quality graphics, or to do quick analysis or design calculations in the context of the complex ensemble.

For individual plots, download graphics for editing in vector graphics programs or download data for local plotting. Alternatively, all job data and plots can be downloaded as a single compressed file.

UTILITIES PAGE

The Utilities page of the NUPACK web app allows users to analyze, design, or annotate the equilibrium properties of a complex. The page accepts as input either sequence information, structure information, or both, performing diverse functions based on the information provided, including:

- Evaluation and display of equilibrium base-pairing information for a specified secondary structure in the context of the complex to which it belongs.
- Automatic layout, rendering, and annotation of secondary structures specified in dot-parens-plus, RLE dot-parens-plus, or DU+ notation.
- Sequence analysis or design for a complex ensemble.

For individual plots, download graphics for editing in vector graphics programs or download data for local plotting. Alternatively, all job data and plots can be downloaded as a single compressed file.

Information can be exported to the Analysis or Design pages:

- *To Analysis*: For the specified complex, export the strand sequences to the Analysis page to analyze in the context of a test tube ensemble.
- *To Design*: For the specified complex, export the specified structure to the Design page to design the sequence in the context of a test tube ensemble, carrying along any specified sequence constraints.

Note that for a given complex ensemble:

- The Analysis page displays results through the lens of the MFE proxy structure.
- The Design page displays results through the lens of the target structure.
- The Utilities page displays results through the lens of a user-specified structure.

METHODS SUMMARY

NUPACK web app frontend. The NUPACK web app frontend is written in Typescript using the React library for user interface logic and Semantic UI for user interface visuals. Bar graphs are generated using Plotly. Structure drawing and concentration calculations are written in C++17 and compiled to Web Assembly to run in the browser.

NUPACK web app control plane. The NUPACK web app control plane is written in Kotlin using PostgreSQL for metadata, AWS S3 for job storage, Redis for caching and communications, and Kubernetes for orchestration of worker containers and scaling clusters.

NUPACK web app backend. The NUPACK 4 backend is written in C++17 using `libsimdpp`³⁰ for SIMD operations, `armadillo`³¹ for linear algebra, `Taskflow`³² for task parallelization, and `gecode`³³ for constraint solving.

NUPACK Python module. The NUPACK 4 Python module is written in C++17 with bindings to Python 3.7+. The `numpy`,³⁴ `scipy`,³⁵ and `pandas`³⁶ packages are used to provide flexible user-friendly numerical interfaces.

RESOURCES

NUPACK web app documentation. Documentation is provided within the web app via the Overview page, the Definitions page, the expandable help text next to each item within the interface, and via the Intros and Demos for the Analysis, Design, and Utilities pages.

NUPACK web app subscriptions. NUPACK is a non-profit academic resource within the Beckman Institute at Caltech. Non-commercial academic users can subscribe to the NUPACK web app to run jobs on the scalable hybrid cloud compute cluster subject to the NUPACK Terms. Commercial users can inquire about obtaining a commercial subscription by contacting info@nupack.org. We are in the process of implementing NUPACK user fees that will help make NUPACK a sustainable resource for the research community.

NUPACK Python module and source code. Non-commercial academic users can download the NUPACK 4 Python module and source code subject to the NUPACK Software License Agreement (nupack.org). Commercial users can inquire about obtaining a commercial license by contacting info@nupack.org. Documentation for the NUPACK 4 Python module is provided via the NUPACK 4 User Guide (docs.nupack.org) including example jobs.

NUPACK technical support. For technical support, feature requests, or bug reports, please contact support@nupack.org.

AUTHOR INFORMATION

Corresponding Author

*E-mail: niles@caltech.edu

Author Contributions

#These authors contributed equally.

Notes

The authors declare the following competing financial interest(s): a patent.

ACKNOWLEDGMENTS

We thank all the NUPACK users that have helped out as alpha and beta testers over the years, as well as the many NUPACK users that have emailed support@nupack.org to request features or report bugs. This work was supported by the National Science Foundation (Software Elements NSF-OAC-1835414, INSPIRE NSF-CHE-1643606, Molecular Programming Project NSF-CCF-1317694, XSEDE NSF-ACI-1548562), by the National Institutes of Health (National Research Service Award T32 GM007616), by a Schmidt Academy Scholarship, by the Caltech cloud credits program from Amazon Web Services, and by the Programmable Molecular Technology Center (PMTTC) within the Beckman Institute at Caltech.

REFERENCES

- (1) Zadeh, J. N., Steenberg, C. D., Bois, J. S., Wolfe, B. R., Pierce, M. B., Khan, A. R., Dirks, R. M., and Pierce, N. A. (2011) NUPACK: Analysis and design of nucleic acid systems. *J. Comput. Chem.* 32, 170–173.
- (2) Dirks, R. M., and Pierce, N. A. (2003) A partition function algorithm for nucleic acid secondary structure including pseudoknots. *J. Comput. Chem.* 24, 1664–1677.
- (3) Dirks, R. M., Lin, M., Winfree, E., and Pierce, N. A. (2004) Paradigms for computational nucleic acid design. *Nucleic Acids Res.* 32, 1392–1403.
- (4) Dirks, R. M., and Pierce, N. A. (2004) An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots. *J. Comput. Chem.* 25, 1295–1304.
- (5) Dirks, R. M., Bois, J. S., Schaeffer, J. M., Winfree, E., and Pierce, N. A. (2007) Thermodynamic analysis of interacting nucleic acid strands. *SIAM Rev.* 49, 65–88.
- (6) Zadeh, J. N., Wolfe, B. R., and Pierce, N. A. (2011) Nucleic acid sequence design via efficient ensemble defect optimization. *J. Comput. Chem.* 32, 439–452.
- (7) Wolfe, B. R., and Pierce, N. A. (2015) Sequence design for a test tube of interacting nucleic acid strands. *ACS Synth. Biol.* 4, 1086–1100.
- (8) Wolfe, B. R., Porubsky, N. J., Zadeh, J. N., Dirks, R. M., and Pierce, N. A. (2017) Constrained multistate sequence design for nucleic acid reaction pathway engineering. *J. Am. Chem. Soc.* 139, 3134–3144.
- (9) Fornace, M. E., Porubsky, N. J., and Pierce, N. A. (2020) A unified dynamic programming framework for the analysis of interacting nucleic acid strands: Enhanced models, scalability, and speed. *ACS Synth. Biol.* 9, 2665–2678.
- (10) Tinoco, I., Jr., Uhlenbeck, O.C., and Levine, M.D. (1971) Estimation of secondary structure in ribonucleic acids. *Nature* 230, 362–367.
- (11) Serra, M. J., and Turner, D. H. (1995) Predicting thermodynamic properties of RNA. *Methods Enzymol.* 259, 242–261.
- (12) SantaLucia, J., Jr. (1998) A Unified View of Polymer, Dumbbell, and Oligonucleotide DNA Nearest-Neighbor Thermodynamics. *Proc. Natl. Acad. Sci. U. S. A.* 95, 1460–1465.
- (13) Xia, T.B., SantaLucia, J., Jr., Burkard, M.E., Kierzek, R., Schroeder, S.J., Jiao, X.Q., Cox, C., and Turner, D.H. (1998) Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry* 37, 14719–14735.
- (14) Mathews, D. H., Sabina, J., Zuker, M., and Turner, D. H. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.* 288, 911–940.
- (15) Bommarito, S., Peyret, N., and SantaLucia, J. (2000) Thermodynamic parameters for DNA sequences with dangling ends. *Nucleic Acids Res.* 28, 1929–1934.
- (16) Peyret, Nicolas, *Prediction of Nucleic Acid Hybridization: Parameters and Algorithms*. Thesis, Wayne State University (2000).
- (17) Zuker, M. (2003) Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* 31, 3406–3415.
- (18) SantaLucia, J., Jr., and Hicks, D. (2004) The thermodynamics of DNA structural motifs. *Annu. Rev. Biophys. Biomol. Struct.* 33, 415–440.
- (19) Mathews, D. H., Disney, M. D., Childs, J. L., Schroeder, S. J., Zuker, M., and Turner, D. H. (2004) Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl. Acad. Sci. U. S. A.* 101, 7287–7292.
- (20) Koehler, R. T., and Peyret, N. (2005) Thermodynamic Properties of DNA Sequences: Characteristic Values for the Human Genome. *Bioinformatics* 21, 3333–3339.
- (21) Lu, Z. J., Turner, D. H., and Mathews, D. H. (2006) A set of nearest neighbor parameters for predicting the enthalpy change of RNA secondary structure formation. *Nucleic Acids Res.* 34, 4912–4924.
- (22) Turner, D. H., and Mathews, D. H. (2010) NNDB: The nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res.* 38, D280–D282.
- (23) Hochrein, L. M., Schwarzkopf, M., Shahgholi, M., Yin, P., and Pierce, N. A. (2013) Conditional Dicer substrate formation via shape and sequence transduction with small conditional RNAs. *J. Am. Chem. Soc.* 135, 17322–17330.
- (24) Hanewich-Hollatz, M. H., Chen, Z., Hochrein, L. M., Huang, J., and Pierce, N. A. (2019) Conditional guide RNAs: Programmable conditional regulation of CRISPR/Cas function in bacterial and mammalian cells via dynamic RNA nanotechnology. *ACS Cent. Sci.* 5, 1241–1249.
- (25) Geary, C., Rothmund, P. W. K., and Andersen, E. S. (2014) A single-stranded architecture for cotranscriptional folding of RNA nanostructures. *Science* 345, 799–804.
- (26) Zadeh, J. N., *Algorithms for Nucleic Acid Sequence Design*. Ph.D. Thesis, California Institute of Technology (2010).
- (27) Bloomfield, V.A., Crothers, D.M., and Tinoco, I., Jr. (2000) *Nucleic Acids: Structures, Properties, and Functions* (University Science Books, Sausalito, CA).
- (28) Porubsky, N. J., *Enhanced Algorithms for Analysis and Design of Nucleic Acid Reaction Pathways*. Ph.D. Thesis, California Institute of Technology (2020).
- (29) Seeman, N. C. (1982) Nucleic acid junctions and lattices. *J. Theor. Biol.* 99, 237–247.
- (30) Kanapickas, P. (2017). Libsimdpp.

<https://github.com/p12tic/libsimdpp>.

- (31) Sanderson, C., and Curtin, R. (2016) Armadillo: A template-based C++ library for linear algebra. *J. Open Source Softw.* 1, 26.
- (32) Huang, T.-W., Lin, D.-L., Lin, C.-X., and Lin, Y. (2021) Taskflow: A lightweight parallel and heterogeneous task graph computing system. *IEEE Trans. Parallel Distrib. Syst.* 33, 1303–1320.
- (33) Gecode Team, (2022). Gecode: Generic Constraint Development Environment. <https://www.gecode.org>.
- (34) Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, Ralf, Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., and et al. (2020) Array programming with NumPy. *Nature* 585, 357–362.
- (35) Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., and et al. (2020) SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* 17, 261–272.
- (36) McKinney, W., and et al. (2011) Pandas: A foundational Python library for data analysis and statistics. *Python High Perform. Sci. Comput.* 14, 1–9.