Biophysical Journal, Volume *117*

Supplemental Information

A Polymer Physics Framework for the Entropy of Arbitrary Pseudoknots

Ofer Kimchi, Tristan Cragnolini, Michael P. Brenner, and Lucy J. Colwell

A polymer physics framework for the entropy of arbitrary pseudoknots Supplementary Information

Ofer Kimchi,[1,*] Tristan Cragnolini,[2] Michael P. Brenner,[3,4] and Lucy J. Colwell[2,†]

[1]*Harvard Graduate Program in Biophysics, Harvard University, Cambridge, MA 02138*
[2]*Department of Chemistry, University of Cambridge, CB2 1EW, Cambridge, United Kingdom*
[3]*School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138*
[4]*Kavli Institute of Bionano Science and Technology, Harvard University, Cambridge, MA 02138*

The supplementary information is divided into several sections. In Sections S1 and S2 we detail our implementation of the nearest-neighbor parameters, as well as the methods used to compare our algorithm's performance to other current models. In Section S3 we discuss how our algorithm can be easily generalized to probe multiple interacting strands including any combination of DNA and RNA. In Section S4 and Fig. S1 we provide a more complete derivation of Eq. 7. In Section S5, we show how to analytically calculate the integrals in Eq. 7. In Section S6 we derive the higher-order corrections to Eq. 5.

In Fig. S2 we display all possible graphs of up to two stems and their respective RNA structures along with the integral formulation of their entropies and their evaluated forms. In Fig. S3 we discuss how our algorithm compares to state-of-the-art prediction tools (the analogue of Fig. 4) when restricting ourselves to structures allowed by the chosen constraints on our algorithm.

In Section S7A we discuss how our algorithm's properties scale with the length of the sequence for random sequences between 10 and 21 ntds in length, shown in Fig. S4. In Section S7B, we provide a mathematical discussion for why the average number of structures for a sequence of length n scales exponentially with n; the discussion corresponds to Fig. S5. We show running time and total number of secondary structure distributions for sequences in our dataset in Fig. S6, with a corresponding discussion in Section S7C.

In Fig. S7, we demonstrate that loop entropies are highly non-negligible; the magnitude of the predicted loop entropy is roughly equal to the magnitude of the total free energy of a structure. In sections S8 and S9 and in figures S8 - S10 we show the entropy calculation for pseudoknots more complex than those in Fig. S2; namely, the kissing hairpins pseudoknot and the most common pseudoknots found in our benchmark dataset. Finally, in Section S10 and Fig. S11, we demonstrate a sample free energy calculation and graph decomposition process.

S1. FURTHER METHOD DETAILS

A. Implementation of nearest-neighbor free energies

Our entropy model (described in the Materials and Methods section) was used in place of the entropies of hairpin, bulge, internal, and multibranch loops and we set the enthalpy terms of these loops (aside from nearest-neighbor interactions) to zero; we did not consider mismatch-mediated coaxial stacking, symmetry penalties or penalties for specific closures of stems; and we implemented coaxial stacking terms in place of terminal mismatches or dangling ends whenever two stems in multibranch loops are directly adjacent.

B. Comparison with other prediction tools

In order to compare the sensitivity and PPV of different prediction tools, we considered the base pairs present in the experimental structure and in each algorithm's MFE structure. Base pairs present in both were labeled as true positives (TP), base pairs present in the predicted algorithm were labeled as false positives (FP) and those present in the experimental structure but not the predicted MFE structure were labeled as false negatives (FN). In order to compare different metrics we use the summary statistics of sensitivity ($TP/TP+FN$) and PPV ($TP/TP+FP$). PPV is a more useful metric for RNA structure prediction algorithms than specificity because the definition of true negatives is unclear when considering base pairs.

*Electronic address: okimchi@g.harvard.edu
†Electronic address: ljc37@cam.ac.uk

The sequences tested were downloaded from the Pseudobase++, RNAStrand, and CompaRNA PDB databases. We constrained database searches to return results only for sequences of length ≤ 80 ntds. We further restricted the search of the RNAStrand database to only include sequences where all nucleotides were known, and to not include fragments, multiple strands, or duplicates. We removed all sequences that had hairpins of under 3 ntds. Finally, we compared the sequence similarity of the sequences derived and kept only sequences with ≥ 0.2 Jukes-Cantor sequence dissimilary measured using the MatLab command `seqpdist`, which aligns sequences using the Needleman-Wunsch algorithm with the *NUC44* scoring matrix. The Jukes-Cantor distance between two sequences is defined as

$$d_{\mathrm{JC}} = -\frac{3}{4}\log\left(1-\frac{4p}{3}\right) \tag{S1}$$

where p is the fraction of sites which differ between the sequences after they have been aligned. By imposing $d_{\mathrm{JC}} \geq 0.2$ we impose a constraint that $p > 0.17$.

We assumed $T = 300K$ for all predictions.

In order to speed up computation for longer sequences, we set the parameter m describing the minimum number of consecutive base pairs in a stem to the minimum value it can take such that the total number of possible stems is less than 150. This latter parameter was chosen arbitrarily and is likely not optimized; however, changing it to 200 had no significant effect (see data in Supplementary Table 1). Setting the maximum total number of possible stems to 150 resulted in $m = 1$ for 22% of the sequences, $m = 2$ for 33% of the sequences, $m = 3$ for 23%, $m = 4$ for 20%, and $m = 5$ for nine sequences. Changing the maximum total number of possible stems to 200 resulted in $m = 1$ for 34% sequences, $m = 2$ for 29% of sequences, $m = 3$ for 22%, $m = 4$ for 15%, and $m = 5$ for one sequence.

Our algorithm can enumerate and calculate the entropies of both parallel and antiparallel stems. (An antiparallel stem is a list of consecutive base pairs of the form $[i \cdot j,\ (i+1)\cdot(j-1),\ (i+2)\cdot(j-2)...]$, while a parallel stem has the form $[i \cdot j,\ (i+1)\cdot(j+1),\ (i+2)\cdot(j+2)...]$.) Parallel stems are disallowed in non-pseudoknotted structures, and are stabilized at certain pH levels. We disallowed parallel stems in our calculations.

As part of the enumeration procedure, we created a compatibility matrix $C_{p,q}$ detailing the compatibility of structures p and q (structures p and q are compatible if they do not share any nucleotides). In practice, since there are some structures whose entropies we have not analytically derived, we found it useful to also construct three- and four-dimensional matrices C_3 and C_4 which define three- and four-way compatibility, in order to exclude most such structures at this stage.

In order to compare topologies, we measure whether the eigenvalue spectra of the two matrices defining the bonds between each node are equal (two matrices are needed because there are two types of bonds). This method is guaranteed to correctly identify graph isomorphisms in all cases but may have false positives. We have found no evidence of false positives in all cases tested (compared against the MatLab `isisomorphic` command).

For the analysis in Fig. 6 we also set $m > 1$ to speed up computation. Starting from the top left and going across, we set $m = (4, 3, 3, 4, 4, 4)$. We also disallowed parallel stems in order to speed up the computation.

S2. PREDICTION TOOL PARAMETERS

To compare our results, we used the implementation of other prediction tools, when provided by the authors. In most cases, program options have been left to their default value. We list below some of the more important options.

- RNAFold: Temperature: 37 C
- Andronescu: Temperature: 37 C
- Mfold: Temperature: 37 C
- CONTRAFold: $\gamma = 6$
- PPfold: N/A
- Centroidfold: $\gamma = 6$
- ContextFold: Model: "trained/StHighCoHigh.model"
- HotKnots DP/RE/CC: energy model DP/RE/CC
- ProbKnot: 1 iteration

- pknots: N/A
- RNAPKplex: Temperature: 37 C
- ILM: N/A

S3. PROBING MULTIPLE INTERACTING STRANDS

The algorithm presented here can also be easily generalized to probe multiple interacting strands, using only one further parameter which has been previously studied to define the free energy cost of forming a duplex [1, 2]. Following Ref. [3] we concatenate the two (or more) sequences, separated by a number of inert nucleotides which serve as a placeholder and which are removed before free energy calculations are implemented.

The algorithm described here can be equally well-applied to DNA strands by using the parameter sets from the SantaLucia laboratory [4] . In addition, our algorithm can probe DNA-RNA bonds using the parameter sets from Refs. [5, 6], and interpolating between the DNA and RNA cases for those parameters that have not yet been tabulated from experimental data. The inclusion of DNA strands may require slight modification to the two entropy parameters (b and v_s) which are based on data from RNA experiments.

S4. DERIVING EQ. 7

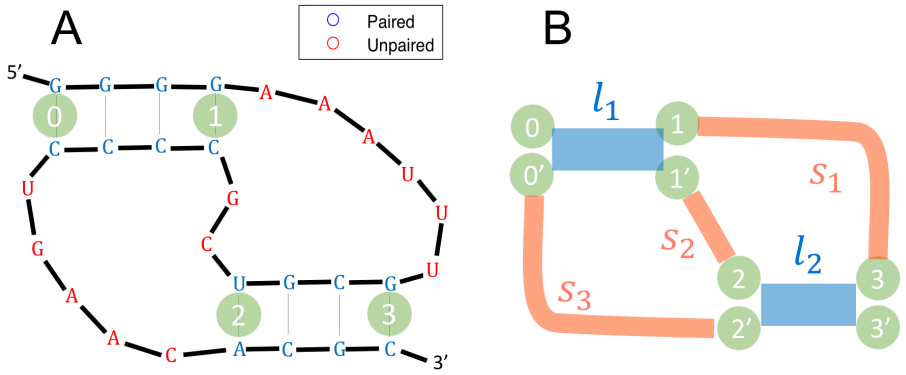

FIG. S1: **A preliminary description of an H-type pseudoknot. A:** An instance of the canonical H-type pseudoknot, reprinted from Fig. 3. **B:** A preliminary version of the graph representing its entropy. In Sec. S4 we demonstrate that this graph is equivalent to that shown in Fig. 3A.

In this section we more fully detail the steps leading to Eq. 7, the entropy of the RNA structure depicted in Fig. S1A.

We start by treating each nucleotide as its own node, subject to the constraint that the distance between nucleotides is given by $a = 0.33$ nm. Writing such an expression is cumbersome, but because of the property of $P_s(\vec{r})$ that $\int P_x(\vec{r_1})P_y(\vec{r_2} - \vec{r_1})d\vec{r_1} = P_{x+y}(\vec{r_2})$, we can simply integrate over all nodes not at the edges of stems.

The full expression for the entropy of this graph is thus given by

$$e^{\Delta S/k_B} = \int d\vec{r_{0'}} \int d\vec{r_1} \int d\vec{r_{1'}} \int d\vec{r_2} \int d\vec{r_{2'}} \int d\vec{r_3} \int d\vec{r_{3'}}\; q(\vec{r_{0'}})\; q(\vec{r_2} - \vec{r_{2'}}) \times$$
$$\delta^3(|\vec{r_1}| - l_1)\; \delta^3(|\vec{r_1} - (\vec{r_{1'}} - \vec{r_{0'}})|)\; \delta^3(|\vec{r_3} - \vec{r_2}| - l_2)\; \delta^3(|\vec{r_3} - \vec{r_2} - (\vec{r_{3'}} - \vec{r_{2'}})|)\; P_{s_1}(\vec{r_3} - \vec{r_1})\; P_{s_2}(\vec{r_2} - \vec{r_{1'}})\; P_{s_3}(\vec{r_{2'}} - \vec{r_{0'}})$$

which is depicted graphically in Fig. S1B. We are using $\delta^3(|x| - a)$ to signify

$$\delta^3(|\vec{x}| - a) = \frac{\delta(|\vec{x}| - a)}{4\pi a^2}; \qquad \int d\vec{x}\; \delta^3(|\vec{x}| - a) = 1. \tag{S2}$$

$\delta^3(|x| - a)$, like $P_s(\vec{r})$, has units of inverse volume.

Vectors are defined relative to the origin where node 0 is placed (i.e. $|\vec{r_0}| = 0$). There is no integration over $\vec{r_0}$ because such an integral would cancel out with the corresponding term in S_{free}, and thus disappear in the formula for ΔS.

$q(\vec{r})$ is defined as the probability of a nucleotide located a vector $\vec{r}$ from the origin to be bonded to a nucleotide located at the origin (assuming the two nucleotides are complementary). If following Ref. [7] we wish to include an upper bound for the bond length, r_s, $q(\vec{r})$ becomes a Heaviside Θ function. Integration over q leads to the definition of v_s: $v_s = \int d\vec{r} q(\vec{r})$.

Only two factors of q are present, as opposed to one factor for each base pair in the structure, because we take the entropy of stems into account separately. For this expression, we treat stems as rigid rods; while the rods have variable and finite width (corresponding to the property that nucleotides do not need to be at a precise separation in order to bond), they cannot be thicker on one end than the other, since including such possibilities would overcount the entropy of the stem. Our expression thereby has the property that it is invariant if we also integrate over two nodes representing two arbitrary base pairs (say, one on the stem between node 0 and node 1, and one between nodes $0'$ and $1'$). The choice of which bonded nodes on each stem to put in the argument of q is arbitrary, but there is only one bonded node (and therefore one q term) for each stem.

We make progress by assuming that because of the q terms and delta functions, nodes representing nucleotides which are bonded are located close enough that the vector $\vec{r}$ between them can be approximated as having zero length within the context of the terms $P_s(\vec{r})$.

We therefore approximate our formula as

$$e^{\Delta S/k_B} = \int d\vec{r_{0'}} \int d\vec{r_1} \int d\vec{r_{1'}} \int d\vec{r_2} \int d\vec{r_{2'}} \int d\vec{r_3} \int d\vec{r_{3'}}\ q(\vec{r_{0'}})\ q(\vec{r_2} - \vec{r_{2'}}) \delta^3(|\vec{r_1} - (\vec{r_{1'}} - \vec{r_{0'}})|) \times$$
$$\delta^3(|\vec{r_3} - \vec{r_2} - (\vec{r_{3'}} - \vec{r_{2'}})|)\ \delta^3(|\vec{r_1}| - l_1)\ \delta^3(|\vec{r_3} - \vec{r_2}| - l_2)\ P_{s_1}(\vec{r_3} - \vec{r_1})\ P_{s_2}(\vec{r_2} - \vec{r_1})\ P_{s_3}(\vec{r_2})$$

By employing transformations as in Section S5 (e.g. $\vec{r'} \equiv \vec{r_{1'}} - \vec{r_{0'}}$), the four integrals over the primed nodes become two integrals over delta functions (which give unity) and two over the q terms. The latter two become two factors of v_s, and we arrive at Eq. 7.

S5. PERFORMING THE GAUSSIAN INTEGRALS

The method of performing the Gaussian integrals of Eq. 7 can be generally applied to the calculation of the entropies of other pseudoknots, and so we describe it in detail here.

Eq. 7 is given by

$$e^{\Delta S/k_B} = v_s^2 \int d\vec{r_1} \int d\vec{r_2} \int d\vec{r_3} \frac{\delta(|\vec{r_1}| - l_1)}{4\pi l_1^2} \frac{\delta(|\vec{r_3} - \vec{r_2}| - l_2)}{4\pi l_2^2} P_{s_1}(\vec{r_3} - \vec{r_1}) P_{s_2}(\vec{r_2} - \vec{r_1}) P_{s_3}(\vec{r_2})$$

We start by utilizing our approximation that the integrals extend over all of space to rewrite $d\vec{r_2}d\vec{r_3}$ as $d\vec{r_2}d(\vec{r_3} - \vec{r_2})$, and we rewrite all instances of $\vec{r_3}$ as $(\vec{r_3} - \vec{r_2}) + \vec{r_2}$.

$$e^{\Delta S/k_B} = v_s^2 \prod_{i=1}^{3} \left(\frac{\gamma}{\pi s_i}\right)^{3/2} \int d\vec{r_1} \frac{\delta(|\vec{r_1}| - l_1)}{4\pi l_1^2} \int d\vec{r_2} \int d(\vec{r_3} - \vec{r_2}) \frac{\delta(|\vec{r_3} - \vec{r_2}| - l_2)}{4\pi l_2^2} \times e^{\gamma\left[-\left(\frac{(\vec{r_3} - \vec{r_2})^2}{s_1}\right) - (\vec{r_2} - \vec{r_1})^2\left(\frac{1}{s_1} + \frac{1}{s_2}\right) - \frac{r_2^2}{s_3} - \frac{2}{s_1}(\vec{r_3} - \vec{r_2}) \cdot (\vec{r_2} - \vec{r_1})\right]},$$

where for notational convenience have defined a parameter $\gamma = 3/2b$.[1].

To do the $(\vec{r_3} - \vec{r_2})$ integral, we convert to polar coordinates such that $(\vec{r_3} - \vec{r_2}) \cdot (\vec{r_2} - \vec{r_1}) = |\vec{r_3} - \vec{r_2}||\vec{r_2} - \vec{r_1}| \cos\theta$. Performing the integral yields

$$e^{\Delta S/k_B} = v_s^2 \prod_{i=1}^{3} \left(\frac{\gamma}{\pi s_i}\right)^{3/2} \frac{e^{-\gamma l_2^2/s_1}}{2} \int d\vec{r_1} \frac{\delta(|\vec{r_1}| - l_1)}{4\pi l_1^2} \int d\vec{r_2}\ e^{\gamma\left[-\frac{r_2^2}{s_3} - (\vec{r_2} - \vec{r_1})^2\left(\frac{1}{s_1} + \frac{1}{s_2}\right)\right]} \left(\frac{e^{(2\gamma l_2 |\vec{r_2} - \vec{r_1}|/s_1)} - e^{(-2\gamma l_2 |\vec{r_2} - \vec{r_1}|/s_1)}}{2\gamma l_2 |\vec{r_2} - \vec{r_1}|/s_1}\right).$$

We now use the same trick from before to rewrite $d\vec{r_2}$ as $d(\vec{r_2} - \vec{r_1})$, and rewrite each instance of $\vec{r_2}$ as $(\vec{r_2} - \vec{r_1}) + \vec{r_1}$. As before, $(\vec{r_2} - \vec{r_1}) \cdot \vec{r_1}$ becomes $|\vec{r_2} - \vec{r_1}||\vec{r_1}| \cos\theta$. Denoting $(\vec{r_2} - \vec{r_1})$ as $\vec{r}$ and doing the integral over r_1 after performing this transformation yields

$$e^{\Delta S/k_B} = v_s^2 \prod_{i=1}^{3} \left(\frac{\gamma}{\pi s_i}\right)^{3/2} \frac{e^{-\gamma\left(\frac{l_2^2}{s_1} + \frac{l_1^2}{s_3}\right)}}{2} \int_0^\infty dr\ r^2\ e^{-\gamma r^2\left(\frac{1}{s_1} + \frac{1}{s_2} + \frac{1}{s_3}\right)} \left(\frac{e^{(2\gamma l_2 r/s_1)} - e^{(-2\gamma l_2 r/s_1)}}{2\gamma l_2 r/s_1}\right) \int_{-1}^{1} d\cos(\theta) e^{-2\gamma \frac{l_1 r}{s_3} \cos(\theta)}.$$

Finally, we perform the integrals remaining to arrive at

$$e^{\Delta S/k_B} = \frac{v_s^2 \gamma^2 \exp\left(-\frac{\gamma\left(l_1^2(s_1+s_2) + l_2^2(s_2+s_3)\right)}{s_1 s_2 + s_1 s_3 + s_2 s_3}\right)}{2\pi^3 l_1 l_2 s_2 \sqrt{s_1 s_2 + s_1 s_3 + s_2 s_3}} \times \sinh\left(\frac{2\gamma l_1 l_2 s_2}{s_1 s_2 + s_1 s_3 + s_2 s_3}\right)$$

where sinh is the hyperbolic sine function. This formula is equivalent to the one presented without proof in Ref. [8].

It can be easily verified (see Fig. S2) that the entropy of an open net can be calculated given the formula for the corresponding closed net, which has an extra single-bond of length s_i, through multiplication by $(\gamma/\pi s_i)^{-3/2}$ and taking the limit $s_i \to \infty$. The formula for the "very open net 2", which is identical to any of the open nets that have two stems after removing the edge corresponding to s_2, can thus be calculated to be

$$e^{\Delta S_{\text{very-open-net-2}}/k_B} = \frac{v_s^2 \gamma^{1/2}}{2\pi^{3/2} l_1 l_2 \sqrt{s_1 + s_2}} \sinh\left(\frac{2\gamma l_1 l_2}{s_1 + s_2}\right) \exp\left(-\gamma \frac{l_1^2 + l_2^2}{s_1 + s_2}\right)$$

where we've labeled the two single-stranded edges' lengths to be s_1 and s_2. This net can form only from two strands binding to one another, as opposed to some of the other nets shown in Fig. S2 which describe two strands bound or one strand with parallel stems.

---

[1] The parameter γ was called β in Refs. [8] and [9]

S6. HIGHER-ORDER CORRECTIONS TO ENTROPY

Eq. 5 , which gives the probability of a random walk of length s to have end-to-end distance $\vec{R}$, is valid only in the limit of $R \gg b$ (where we've denoted $R \equiv |\vec{R}|$). For shorter walks, the Central Limit Theorem no longer holds. In this section, we show a systematic approach to deriving higher-order corrections to the probability distribution given by Eq. 5. The approach taken here is based on a textbook by Ariel Amir (to be published).

We consider n steps in three dimensions, where each step is taken to be of length b with equal probabilities in all directions. Thus, $s = nb$. The probability distribution for where a walker will be after $n = 1$ steps is given by $P_{n=1}(\vec{R}) \equiv \delta(|R| - b)/4\pi b^2$. After two steps, the probability distribution for where the walker will be is given by

$$P_2(\vec{R}) = \int d\vec{R_1} P_1(\vec{R_1}) P_1(\vec{R} - \vec{R_1}). \tag{S3}$$

The form of Eq. S3 is that of a convolution of $P_1(\vec{R})$ with itself. In order to iterate many convolutions easily, we move to Fourier space, since the Fourier transform of a convolution is the product of Fourier transforms. Fourier transforming $P_1(\vec{R})$ yields its characteristic function: $\hat{p}_1(\vec{\omega}) = \int\int\int_{-\infty}^{\infty} d\vec{R}\ P_1(\vec{R}) e^{i\vec{\omega}\cdot\vec{R}}$, which simplifies to

$$\hat{p}_1(\omega) = \frac{\sin(\omega b)}{\omega b} \tag{S4}$$

which only depends on $\omega \equiv |\vec{\omega}|$.

In order to iterate n convolutions in real space, we can simply take the n^{th} power of the Fourier transform, finding

$$\hat{p}_n(\omega) = \left(\sin(\omega b)/\omega b\right)^n . \tag{S5}$$

Taking the inverse Fourier transform, we find

$$P_n(\vec{R}) = \frac{2}{(2\pi)^2} \int_0^{\infty} d\omega\ \omega^2 \left(\frac{\sin(\omega b)}{\omega b}\right)^n \frac{\sin(\omega R)}{\omega R}. \tag{S6}$$

At this point, we use our assumption that n is large. This formula tends to zero for large values of ωb, and we therefore Taylor expand the sin function for small ωb. If we take only the first two terms of this series, we would arrive at Eq. 5 ; we therefore take the first three terms to get the first correction to Eq. 5 . Higher-order corrections can be found by simply taking more terms of the series. Eq. S6 thus becomes

$$P_n(\vec{R}) = \frac{2}{(2\pi)^2} \int_0^{\infty} d\omega\ \omega^2 e^{n \log\left(1 - \frac{(\omega b)^2}{6} + \frac{(\omega b)^4}{120} + \mathcal{O}(\omega b)^6\right)} \frac{\sin(\omega R)}{\omega R}$$

Next, we Taylor expand the logarithm and write the sin as a sum of exponentials. Since the two terms in the sum are identical under the exchange $\omega \to -\omega$, we combine them into one term by changing the lower limit of integration to $-\infty$.

$$P_n(\vec{R}) = \frac{1}{(2\pi)^2 iR} \int_{-\infty}^{\infty} d\omega\ \omega e^{-n\left[\frac{(\omega b)^2}{6} + \frac{(\omega b)^4}{180} + \mathcal{O}(\omega b)^6\right] + i\omega R}. \tag{S7}$$

If we didn't have the quartic term, this integral would be Gaussian and would result in Eq. 5. However, if we keep this term, the integral is no longer solvable analytically. We proceed by setting

$$e^{-n\left[\frac{(\omega b)^4}{180}\right]} = 1 - \frac{n(\omega b)^4}{180} + \mathcal{O}(\omega b)^8. \tag{S8}$$

As is apparent, the finite truncation of this series results in corrections of higher order than the truncation of the series for $\sin(\omega b)$ or of the logarithm above.

Using this series expansion, Eq. S7 becomes a Gaussian integral, which can be solved analytically to yield

$$P_n(\vec{R}) = \left(\frac{3}{2\pi sb}\right)^{3/2} e^{\left(-\frac{3R^2}{2sb}\right)} \left[1 - \frac{3\left(5s^2b^2 - 10sbR^2 + 3R^4\right)}{20s^3b}\right]. \tag{S9}$$

where we've replaced n by s/b.

One of the essential properties of $P_n(\vec{R})$ for our formalism to function is that $\int P_{n_1}(\vec{R}_1)P_{n_2}(\vec{R}_2 - \vec{R}_1)d\vec{R}_1 = P_{n_1+n_2}(\vec{R}_2)$. One can check directly that this holds for Eq. S9. Keeping only first-order correction terms, and defining $\vec{R}_{21} = \vec{R}_2 - \vec{R}_1$,

$$\begin{aligned}
&\int P_{n_1}(\vec{R}_1)P_{n_2}(\vec{R}_2 - \vec{R}_1)d\vec{R}_1 \\
&= \int d\vec{R}_1 \left(\frac{3^2}{2^2\pi s_1 s_2 b^2}\right)^{3/2} e^{\left[-\frac{3}{2b}\left(\frac{R_1^2}{s_1} + \frac{\vec{R}_{21}^2}{s_2}\right)\right]} \left[1 - \frac{3\left(5s_1^2b^2 - 10s_1bR_1^2 + 3R_1^4\right)}{20s_1^3b} - \frac{3\left(5s_2^2b^2 - 10s_2b\vec{R}_{21}^2 + 3\vec{R}_{21}^4\right)}{20s_2^3b}\right] \\
&= \left(\frac{3}{2\pi(s_1+s_2)b}\right)^{3/2} e^{\left(-\frac{3R_2^2}{2(s_1+s_2)b}\right)} \left[1 - \frac{3\left(5(s_1+s_2)^2b^2 - 10(s_1+s_2)bR_2^2 + 3R_2^4\right)}{20(s_1+s_2)^3b}\right] = P_{n_1+n_2}(\vec{R}_2).
\end{aligned}$$

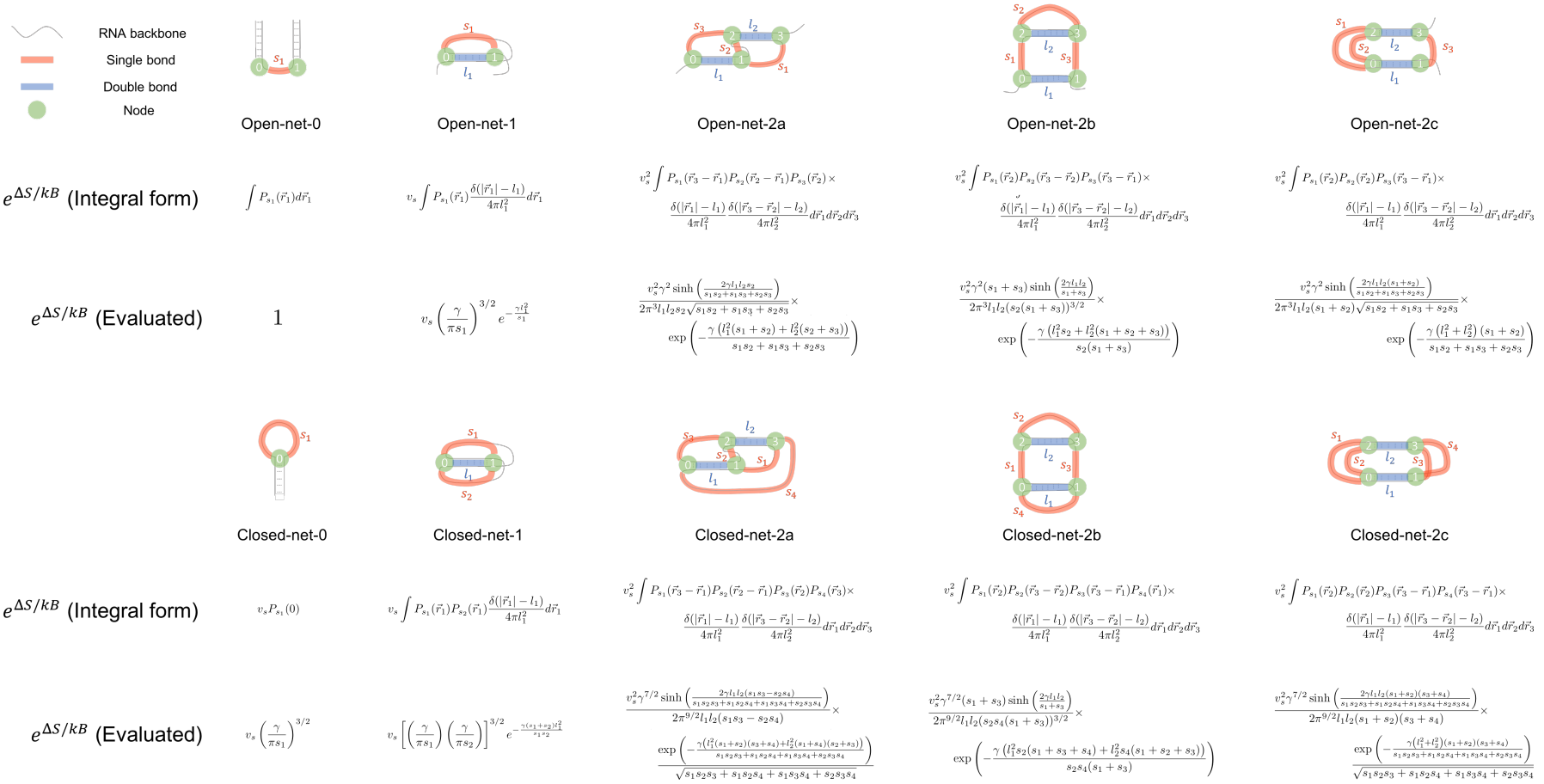

FIG. S2: **Graphs of simple RNA structures.** The 10 graphs with at most two regions of double-stranded RNA and their corresponding RNA backbones are displayed alongside integral and evaluated expressions for the entropy of each graph. Note that stems shown as parallel could be antiparallel if the system considered is comprised of more than one strand. For example, closed-net-1 is an *inter*molecular kissing hairpin complex. We do not include the "very open net 2", a bi-molecular structure that can be created by taking any of the open nets with two stems and removing the edge corresponding to s_2; the relevant calculation is described in the main text. See Fig. 2 of Ref. [9] for comparison. See Section S5 for a description of how to perform the integrals, and for a discussion of how to easily calculate the entropy of an open-net given that of the corresponding closed-net.

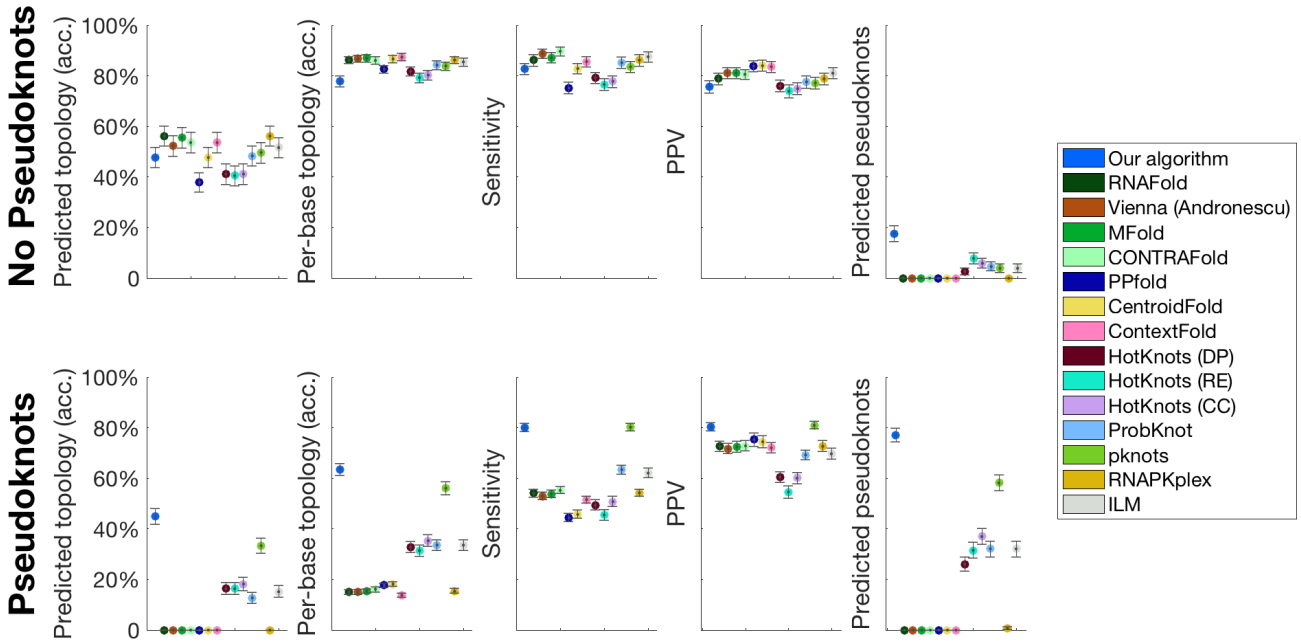

FIG. S3: **Results only including sequences whose structure our algorithm could have predicted.** We consider only the 153 non-pseudoknotted and 165 pseudoknotted sequences whose structures do not include base pairs or topologies disallowed by our algorithm. In this case, we predict the correct topology with 49% (47%) accuracy for non-pseudoknotted (pseudoknotted) structures. This number increases to 62% (82%) and 67% (85%) for top-5 and top-10 accuracy. Surprisingly, we therefore find that our algorithm actually performs better in predicting the pseudoknotted structures in the databases used than the non-pseudoknotted structures. The main results are the same for this dataset as for the full dataset plotted in Fig. 4: our algorithm outperforms all 14 algorithms tested against in predicting pseudoknotted structures, and performs on par with the other algorithms in predicting non-pseudoknotted structures, even though it uses orders of magnitude fewer entropic parameters than the other algorithms tested against.

The constraint placed on allowed sequences in this figure allows us to address to what extent the polymer physics entropy model developed in this work is responsible for our good results, rather than the enumeration scheme. This figure represents a control of the enumeration procedure; pknots, which comes closest to our algorithm's success, only predicts seven sequences included in this dataset to fold into a structure more complex than our algorithm's chosen constraints allow. Removing these seven sequences (in addition to those already removed) does not have a significant effect on the results presented in this figure. (The largest effect is in the accuracy of the predicted topology which increases for pknots from 0.33 to 0.35). We conclude that the difference between our novel entropy model and pknots' (or other algorithms') phenomenological model, rather than the difference in the enumeration procedure, is primarily responsible for the success of our algorithm compared to current metrics.

S7. SCALING OF THE ALGORITHM PROPERTIES FOR RANDOM SEQUENCES AND DISTRIBUTIONS OF ALGORITHM PROPERTIES FOR SEQUENCES IN THE BENCHMARK DATASET

A. Scaling for random sequences

In order to test the scaling properties of the algorithm, we input 100 random sequences for each length between 10 and 21 nucleotides, and set $m = 1$. We plot various properties of the results as a function of the length of the sequence in Fig. S4A. Blue circles are datapoints for each of the 100 sequences in each column. Purple points show the mean. The number of secondary structures grows exponentially with the length of the sequence, as expected due to the brute-force nature of the algorithm, though the number of possible stems grows sub-exponentially. These results are explained later in this section. Similarly, the number of topologies grows exponentially. The probability of forming a pseudoknot appears to plateau at around 10%.

In Fig. S4B, we show that the time the algorithm takes to calculate free energies (the rate limiting step for sequences of any substantial length) grows approximately linearly with the number of possible secondary structures. This is precisely as expected, since the algorithm independently calculates the free energy of each structure, in a process that is easily parallelizable. Deviations from linearity are presumably due to memory constraints which lead to increased computational time for sequences for which many structures need to be stored. While it is customary to plot the time taken as a function of sequence length, as shown in panel A there is a wide variability for each sequence length in the total number of structures, and therefore a similarly wide variability in the time taken. The time taken by the algorithm for a given sequence is better-predicted by the total number of structures enumerated for that sequence than by its length. As shown in panel A (top left) and explained below, the average number of structures for a given sequence grows exponentially with the sequence length, and therefore, the total time taken by the algorithm also grows exponentially with sequence length.

In panel C, we show that for large numbers of stems, the number of possible secondary structures grows as a power law with the number of possible stems. This sub-exponential behavior is due to the fact that some stems cannot coexist in the same structure (if they share any of the same nucleotides or if their coexistence leads to a topology more complex than those in Fig. S2).

B. Scaling of number of structures with sequence length

One main result of the above analysis is that the algorithm runtime is dominated by the scaling properties of the number of structures with the length of the sequence. We therefore sought to better understand this scaling, especially for longer sequences which are not examined in Fig. S4.

A first-order estimate ignores the steric effects of pairing (such as the constraint that if two nucleotides are within a certain linear distance in sequence space, they cannot pair to one another as doing so would create a hairpin that is too small). We make this approximation, and only consider that two nucleotides can pair if they are complementary, and importantly, cannot pair to more than one partner within the same structure. The neglected effect is of course important, though it is expected to give only a higher order correction (i.e. it will not be the dominant effect for purposes of examining scaling behavior for long sequences). The exception of course is for short sequences for which steric effects will be significant – and for which we have enumerated a representative sample of possible structures in Fig. S4. If sterics have any significant effect, it will be to decrease the number of possible structures for short sequences especially, and the effect will be less pronounced for longer sequences which are those we are concerned with here.

For each sequence of length n, we can therefore make structures that include up to $j_{\max} = \text{floor}(n/2)$ base pairs. We can enumerate the number of structures with j base pairs, which we call $N^j_{\text{structures}}$, and then sum this function up for j values from 1 to $j_{\max}$. In other words

$$N^{\text{total}}_{\text{structures}} = \sum_{j=1}^{j_{\max}} N^j_{\text{structures}}$$

We calculate $N^j_{\text{structures}}$ by going base pair by base pair. For the first base pair, there are n first nucleotides to choose from, and on average $3(n-1)/8$ complementary nucleotides. Since we could flip which nt is chosen first and which second, we also multiply by a factor of 1/2. Once the first base pair is chosen, there are $n-2$ nts remaining, which form a sequence of length $n-2$ nts which can be analyzed just as the previous sequence of length n (in other

words, we're describing a mathematically recursive process). Finally, for a structure comprised of j base pairs, there are $j!$ possible (equivalent) orderings of base pairs. Therefore,

$$N^{j}_{\text{structures}} = \frac{1}{j!}\prod_{i=0}^{j-1}\frac{3}{16}(n-2i)(n-2i-1). = \frac{1}{j!}\left(\frac{3}{16}\right)^{j}\frac{n!}{(n-2j)!}.$$

Simplifying, we find that the average total number of structures for a sequence of length n, ignoring steric constraints, is

$$N^{\text{total}}_{\text{structures}} = \sum_{j=1}^{\text{floor}(n/2)}\frac{1}{j!}\left(\frac{3}{16}\right)^{j}\frac{n!}{(n-2j)!}.$$

We tested this equation by explicitly enumerating all possible sequences of length up to 20 nucleotides (see Python code posted to GitHub) and finding perfect agreement with the equation.

This result demonstrates that the total number of structures grows approximately exponentially with the length of the sequence, even for sequences much longer than those examined in Fig. S4 (for which this exponential scaling was also apparent). We plot the result for sequences up to length 400 in Fig. S5. Despite slight curvature for short sequences (for which this naive scaling estimate will not be accurate since steric constraints will be dominant), the result shows exponential growth of the total number of possible structures with the length of the sequence.

As the figure makes clear, the number of possible structures places a significant limit on the length of sequences one can consider by complete landscape enumeration. However, the limit is not nearly as bad as what is suggested by the figure, since the steric considerations ignored to produce it eliminate many structures. Furthermore, by considering stems of length m rather than single base pairs, we can reach sequences up to around 90 nts.

C. Distribution of algorithm scaling properties for benchmark dataset sequences

In Fig. S6 we describe running time and secondary structure count distributions for sequences in the benchmark dataset. We show the histogram of the total time taken to run the algorithm with $N^{\text{max}}_{\text{stems}} = 150$ in panel A (left), finding that the longest time taken was 25 minutes for one sequence. In panel A (middle) we show a histogram of the total number of secondary structures enumerated by the algorithm, finding a wide distribution spanning several orders of magnitude. We also demonstrate that we are in the regime where parallelization will strongly affect the runtime of the algorithm by showing (panel A, right) that the free energy calculation took several times longer than the enumeration procedure. We note however that the details of this calculation (especially the graph decomposition procedure which takes the bulk of the time) have likely not been optimized.

In panel B we show similar plots for the case when no constraints on the types of pseudoknots possible were included (i.e. pseudoknots more complex than those shown in Fig. S2 were also enumerated). We show that including these pseudoknots increases the time it takes to enumerate the structures significantly; the maximum time for a single sequence using $N^{\text{max}}_{\text{stems}} = 150$ increases to 11 hours (panel B, left). While the total number of enumerated secondary structures also increases dramatically (panel B, middle) by leveraging the parallelizability of the algorithm we remain well within the realm of feasibility given the rapid recent growth of available computing power. We also demonstrate (panel B, right) that by decreasing $N^{\text{max}}_{\text{stems}}$ even to 100, orders of magnitude fewer structures are enumerated. The time taken to enumerate the structures also decreases significantly (the maximum is 9 minutes). Our results demonstrate that even exponential-time algorithms such as this complete enumeration are not prohibitive.

In Fig. S7 we examine the loop entropies for the MFE structures predicted by our algorithm for the sequences in the benchmark dataset. We find that the loop entropy (multiplied by temperature) ranges from 5-35 kcal/mol, and is in particular higher for pseudoknotted structures. We further find that the magnitude of the loop entropy is on average slightly over half that of the stem free energy, but represents a higher fraction for pseudoknotted structures. Since the loop entropies contribute in opposite sign to the stem free energies, this demonstrates that as a general rule, the magnitude of the predicted loop entropy is roughly equal to the magnitude of the total free energy of a structure. The accuracy of the loop entropy model is therefore highly significant.

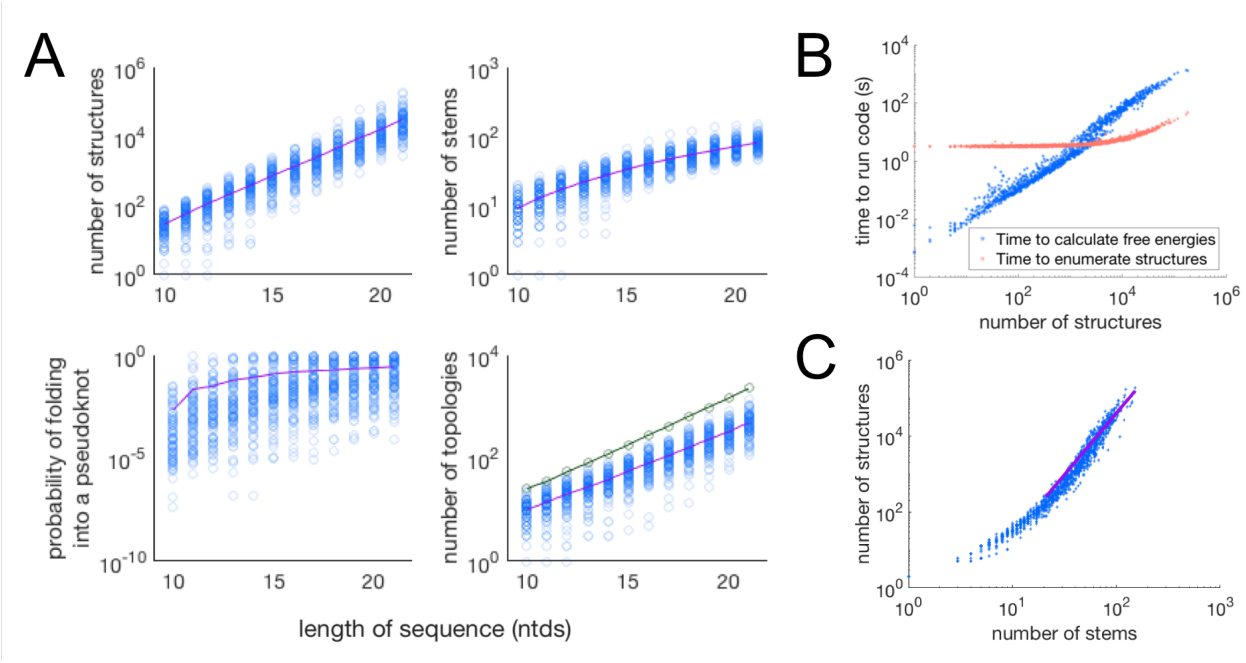

FIG. S4: **Scaling of the algorithm properties with length of sequence.** We input 100 random sequences for each length between 10 and 21 nucleotides into the algorithm. **(A)** Various properties of the results are plotted as a function of the length of the sequence. Blue circles are datapoints for each of the 100 sequences in each column. Purple points show the mean. The number of secondary structures grows exponentially with the length of the sequence, as expected due to the brute-force nature of the algorithm, though the number of possible stems grows sub-exponentially. The probability of forming a pseudoknot appears to plateau at around 10%. The number of topologies grows exponentially (we exclude topologies more complex than those shown in Fig. S2 and the structures leading to them). The green line shows the total number of different topologies over all 100 sequences of a given length. We disallowed parallel stems for this analysis. **(B)** The time the algorithm takes to calculate free energies grows approximately linearly with the number of possible secondary structures, and therefore exponentially with sequence length (see panel A, top left). The data is well-fit to a power law $y = ax^b$ with parameters $a = (3.8 \pm 0.3) * 10^{-4}$ and $b = 1.27 \pm 0.01$. The time taken to enumerate all the structures is constant for short sequences (when few structures are enumerated and the algorithm's overhead is the rate-limiting factor) and then grows as a power law. For sequences of any substantial length, the algorithm is rate-limited by the time it takes to compute free energies, rather than the time taken to enumerate structures. The MatLab program was run on a MacBook Pro 2012 laptop with a 2.3 GHz Intel Core i7 processor and 8 GB memory. **(C)** For large numbers of stems, the number of possible secondary structures grows as a power law with the number of possible stems. This sub-exponential behavior is because some stems cannot coexist in the same structure (if they share any of the same nucleotides or if their coexistence leads to a topology more complex than those in Fig. S2). The purple line shows a fit to the equation $y = ax^b$ with $R^2 = 0.81$. The best-fit values of a and b are found to be $a = 0.0129 \pm 0.0065$ and $b = 3.24 \pm 0.11$.

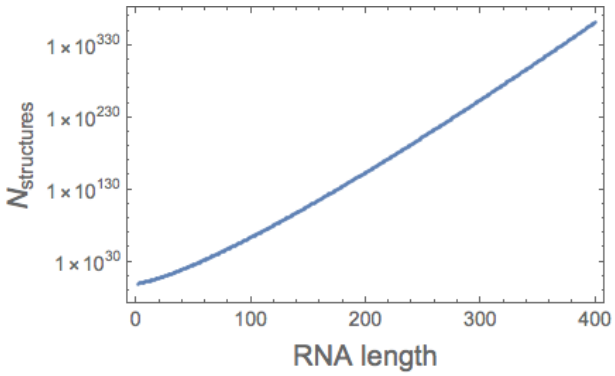

FIG. S5: **Simple scaling estimate for the total number of structures with sequence length.** In Section S7 we find an exact formula for the average number of possible structures as a function of sequence length, neglecting steric effects. Here we plot the results of that formula. We find that despite slight curvature for short sequences (for which this naive scaling estimate will not be accurate since steric constraints will be dominant), the result shows exponential growth of the total number of possible structures with the length of the sequence. Plot created using Mathematica

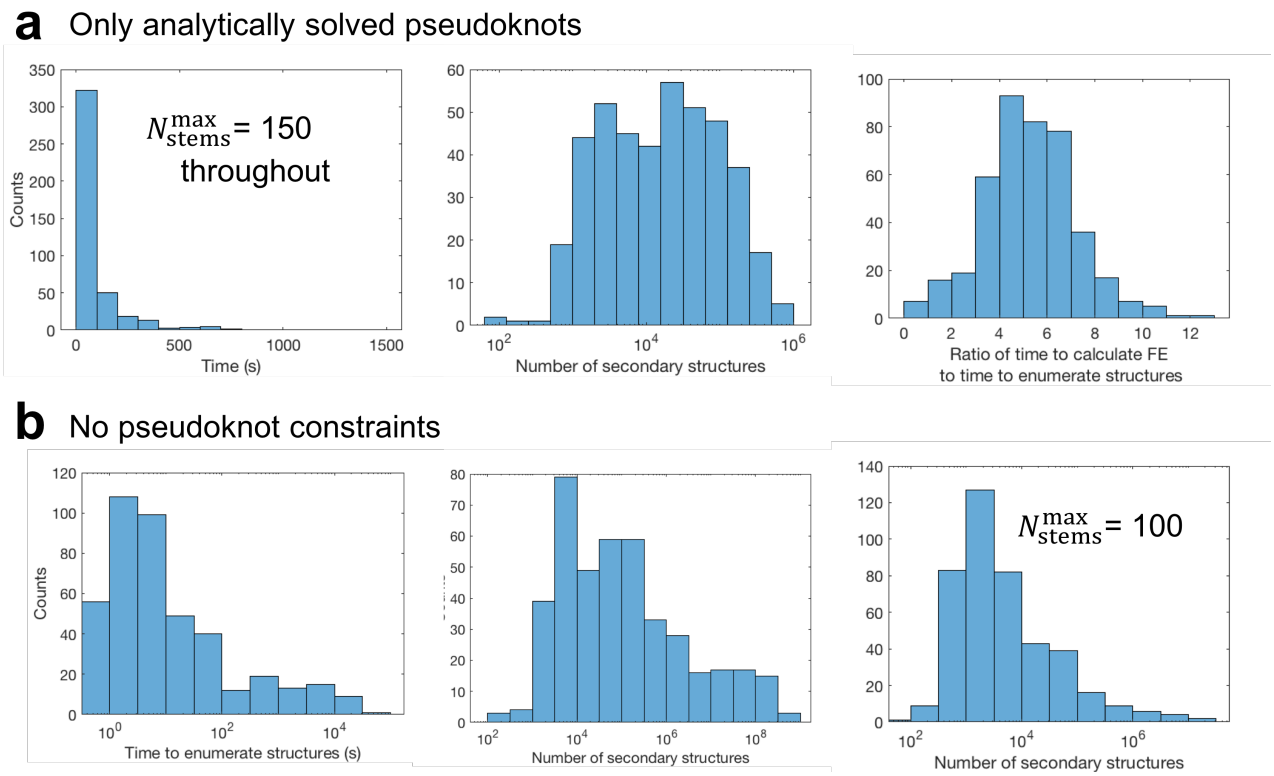

FIG. S6: **Running time and secondary structure count distributions for sequences in the benchmark dataset. a: Left:** Histogram of the total time taken to run the algorithm with $N_{\text{stems}}^{\max} = 150$ for sequences in the benchmark dataset. The longest time taken was 25 minutes for one sequence. Unlike Fig. S4, these results were calculated on a Macbook Pro 2016 laptop with a 3.1 GHz Intel Core i7 processor and 16 GB memory. **Middle:** A histogram of the total number of secondary structures enumerated by the algorithm. **Right:** Calculating the free energy (FE) took several times longer than the enumeration procedure, though the details of this calculation (especially the graph decomposition procedure which takes the bulk of the time) have likely not been optimized. **b:** Results when no constraints on the types of pseudoknots possible were included (i.e. pseudoknots more complex than those shown in Fig. S2 were also enumerated) **Left:** Including all types of pseudoknots increases the time it takes to enumerate the structures significantly; the maximum time for a single sequence using $N_{\text{stems}}^{\max} = 150$ increases to 11 hours. **Middle:** The total number of enumerated secondary structures also increases dramatically, but remains well within the realm of feasibility given the rapid recent growth of available computing power. **Right:** Orders of magnitude fewer structures are enumerated if $N_{\text{stems}}^{\max}$ is decreased even to 100. The time taken to enumerate the structures also decreases significantly (the maximum is 9 minutes). Our results demonstrate that even exponential-time algorithms such as this complete enumeration are not prohibitive.

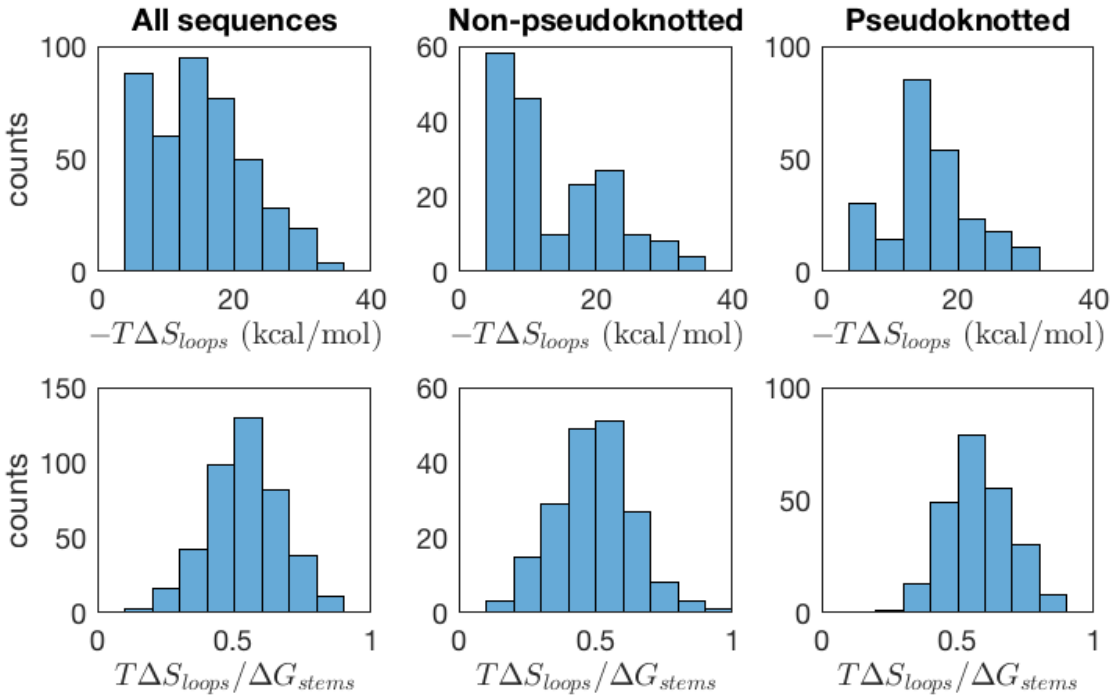

FIG. S7: **Loop entropy statistics.** We examine the loop entropies for the MFE structures predicted by our algorithm for the sequences in the benchmark dataset. We show the results for all sequences (first column), only non-pseudoknotted structures (second column), and only pseudoknotted structures (third column). We considered the predicted structures for the purposes of this classification, but the results don't change significantly if they are classified based on the experimental structures. **The first row** shows the magnitude of the predicted loop entropies. We find that the loop entropies range from 0 to ∼35 kcal/mol, and are in particular higher for pseudoknotted structures, as expected. **The second row** shows the ratio between the magnitude of the predicted loop entropies and the stem free energies $\Delta G_{\text{stems}} = \Delta H_{\text{stems}} - T\Delta S_{\text{stems}}$, which were calculated using the Turner parameters. We find that the magnitude of the loop entropy is on average half that of the stem free energy, but represents a higher fraction for pseudoknotted structures. Since the loop entropies contribute in opposite sign to the stem free energies, this demonstrates that as a general rule, the magnitude of the predicted loop entropy is roughly equal to the magnitude of the total free energy of a structure.

S8. APPLYING OUR FORMALISM TO KISSING HAIRPIN PSEUDOKNOTS

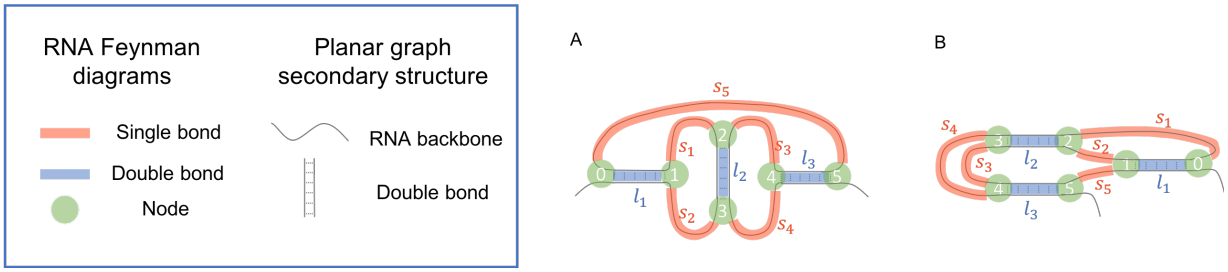

FIG. S8: **Examples of topologies whose entropies need to be solved numerically. A** A kissing hairpin pseudoknot. **B** The most common topology in the dataset which is more complex than those allowed by our chosen constraints. It is equivalent to an H-type pseudoknot with an internal loop in one of the stems.

A biologically common complex pseudoknot for which no entropy calculation has been available is the kissing-hairpin pseudoknot (Fig. S8A). Using our formalism, the entropy of this structure can be estimated by solving the integral

$$e^{\Delta S/k_B} = v_s^3 \int d\vec{r_1} \int d\vec{r_2} \int d\vec{r_3} \int d\vec{r_4} \int d\vec{r_5} \; \frac{\delta(|\vec{r_1}| - l_1)}{4\pi l_1^2} \frac{\delta(|\vec{r_3} - \vec{r_2}| - l_2)}{4\pi l_2^2} \frac{\delta(|\vec{r_5} - \vec{r_4}| - l_3)}{4\pi l_3^2} \times$$
$$P_{s_1}(\vec{r_5}) P_{s_2}(\vec{r_2} - \vec{r_1}) P_{s_3}(\vec{r_3} - \vec{r_1}) P_{s_4}(\vec{r_4} - \vec{r_2}) P_{s_5}(\vec{r_4} - \vec{r_3}). \quad \text{(S10)}$$

We describe the process by which this integral can be solved. First, we let $\gamma = 3/2b$ and call $s_i' = s_i/\gamma$, neglecting the primes from here on for notational convenience. We let $\alpha = \pi^{-15/2}(s_1 s_2 s_3 s_4 s_5)^{-3/2}(v_s/4\pi)^3 (l_1 l_2 l_3)^{-2}$. We let $\vec{r_{ij}} = \vec{r_i} - \vec{r_j}$. The main difficulty in solving these integrals is choosing the proper integration variables. The integral is

$$e^{\Delta S/k_B} = \alpha \int d\vec{r_{54}}\; d\vec{r_4}\; d\vec{r_{32}}\; d\vec{r_2}\; d\vec{r_1}\; \delta(|\vec{r_1}| - l_1)\; \delta(|\vec{r_{32}}| - l_2)\; \delta(|\vec{r_{54}}| - l_3) \times$$
$$\exp\left[-(\vec{r_{54}} + \vec{r_4})^2/s_5 - (\vec{r_2} - \vec{r_1})^2/s_1 - (\vec{r_{32}} + \vec{r_2} - \vec{r_1})^2/s_2 - (\vec{r_4} - \vec{r_2})^2/s_3 - (\vec{r_4} - \vec{r_{32}} - \vec{r_2})^2/s_4\right]. \quad \text{(S11)}$$

We can now proceed to first do the $\vec{r_{54}}$ integral, following the same procedure as in Section S5. $\vec{r_{54}} \cdot \vec{r_4}$ becomes $|\vec{r_{54}}||\vec{r_4}| \cos\theta$ where θ is the integral between the vectors $\vec{r_{54}}$ and $\vec{r_4}$. The integral over all terms containing $\vec{r_{54}}$ yields $\frac{\pi l_3 s_5}{r_4} e^{-l_3^2/s_5 - r_4^2/s_5}(e^{2l_3 r_4/s_5} - e^{-2l_3 r_4/s_5})$.

We can similarly do the $\vec{r_1}$ integral. In order to do so, we define a variable $x = \vec{r_2}(1/s_1 + 1/s_2) + \vec{r_{32}}/s_2$. Thus, $\vec{r_1}$ only appears in our integrals as r_1^2 and as $\vec{r_1} \cdot x$. In order to change the integration variable r_2 to x, we need to introduce the Jacobian $J = (s_1 s_2/s_1 + s_2)^3$. We also set $a = \frac{s_1}{s_3} - \frac{s_2}{s_4}$. After doing the integral, we can expand out the exponent to get

$$e^{\Delta S/k_B} = \alpha J \pi^2 l_1 l_3 s_5 e^{-l_3^2/s_5 - l_1^2(\frac{1}{s_1} + \frac{1}{s_2})} \int d\vec{r_4} \frac{1}{r_4} e^{-r_4^2(\frac{1}{s_3} + \frac{1}{s_4} + \frac{1}{s_5})} \left(e^{2l_3 r_4/s_5} - e^{-2l_3 r_4/s_5}\right) \times$$
$$\int d\vec{x} \frac{1}{x} e^{-x^2\left[\frac{s_1 s_2}{s_1+s_2} + (\frac{s_1 s_2}{s_1+s_2})^2(\frac{1}{s_3} + \frac{1}{s_4})\right]} \left(e^{2l_1 x} - e^{-2l_1 x}\right) e^{2\vec{x}\cdot\vec{r_4}\left(\frac{s_1 s_2}{s_1+s_2}(\frac{1}{s_3} + \frac{1}{s_4})\right)} \times$$
$$\int d\vec{r_{32}} \delta(|\vec{r_{32}}| - l_2) e^{-r_{32}^2\left(\frac{1}{s_1+s_2} + \frac{s_1^2/s_3 + s_2^2/s_4}{(s_1+s_2)^2}\right)} \; e^{2\vec{x}\cdot\vec{r_{32}}\frac{s_1 s_2 a}{(s_1+s_2)^2} - 2\vec{r_4}\cdot\vec{r_{32}}\frac{a}{s_1+s_2}}. \quad \text{(S12)}$$

As can be seen, if $a = 0$, meaning $\frac{s_1}{s_3} = \frac{s_2}{s_4}$, then $\vec{r_{32}}$ only enters our equations as r_{32}^2. In this case, integration over $\vec{r_{32}}$ simply yields $4\pi l_2^2 e^{-l_2^2\left(\frac{1}{s_1+s_2} + \frac{s_1^2/s_3 + s_2^2/s_4}{(s_1+s_2)^2}\right)}$. Setting θ to be the angle between $\vec{r_4}$ and $\vec{x}$, integration over θ proceeds as in previous cases. Integration over the remaining three angles gives $8\pi^2$. Thus,

$$e^{\Delta S/k_B}(a=0) = \alpha J 16\pi^5 l_1 l_2^2 l_3 s_5 \left(\frac{s_1+s_2}{s_1 s_2(\frac{1}{s_3}+\frac{1}{s_4})}\right) e^{-l_1^2(\frac{1}{s_1}+\frac{1}{s_2})-l_2^2\left(\frac{1}{s_1+s_2}+\frac{s_1^2/s_3+s_2^2/s_4}{(s_1+s_2)^2}\right)-l_3^2/s_5} \times$$
$$\int_0^\infty dr_4 e^{-r_4^2(\frac{1}{s_3}+\frac{1}{s_4}+\frac{1}{s_5})}\left(e^{2l_3 r_4/s_5}-e^{-2l_3 r_4/s_5}\right)\int_0^\infty dx e^{-x^2\left[\frac{s_1 s_2}{s_1+s_2}+(\frac{s_1 s_2}{s_1+s_2})^2(\frac{1}{s_3}+\frac{1}{s_4})\right]}\left(e^{2l_1 x}-e^{-2l_1 x}\right)\times$$
$$\left(e^{2xr_4\left(\frac{s_1 s_2}{s_1+s_2}(\frac{1}{s_3}+\frac{1}{s_4})\right)}-e^{-2xr_4\left(\frac{s_1 s_2}{s_1+s_2}(\frac{1}{s_3}+\frac{1}{s_4})\right)}\right). \quad \text{(S13)}$$

These integrals can be solved analytically (by completing the square in the exponent for each of the eight terms in the sum). The result is

$$e^{\Delta S/k_B}(a=0) = \alpha J 16\pi^5 l_1 l_2^2 l_3 s_5 \left(\frac{s_3 s_4(s_1+s_2)}{s_1 s_2(s_3+s_4)}\right) e^{-l_1^2(\frac{s_1+s_2}{s_1 s_2})-l_2^2\left(\frac{s_{1234}+s_3 s_4(s_1-s_2)a}{s_3 s_4(s_1+s_2)^2}\right)-l_3^2/s_5} \times$$
$$2\pi(s_1+s_2)\sqrt{\frac{s_3 s_4 s_5}{s_1 s_2(s_q+s_{1234})}}\, e^{\frac{l_1^2 s_3 s_4(s_1+s_2)^2}{s_1 s_2 s_{1234}}+\frac{l_1^2 s_q^2+l_3^2 s_{1234}^2}{s_5 s_{1234}(s_q+s_{1234})}} \sinh\left(\frac{2l_1 l_3 s_q}{s_5(s_q+s_{1234})}\right) \quad \text{(S14)}$$

where we've defined $s_q = s_5(s_1+s_2)(s_3+s_4)$ and $s_{1234} = s_1 s_2 s_3 + s_1 s_2 s_4 + s_1 s_3 s_4 + s_2 s_3 s_4$. We've written the solution so that the bottom line is the result of the integrals, and written the prefactor of l_2^2 in a way that clarifies how it simplifies.

We can simplify the final result by introducing the variables

$$s_A = \frac{s_5(s_q+s_{1234})}{s_q}; \quad s_B = \frac{s_3 s_4(s_1+s_2)^2}{s_{1234}}; \quad s_v = \sqrt{s_q+s_{1234}}.$$

yielding

$$e^{\Delta S/k_B}(a=0) = \frac{(v_s/s_v)^3}{2\pi^{9/2}}\frac{s_A}{l_1 l_3} e^{-\left(\frac{l_1^2+l_3^2}{s_A}\right)-\frac{l_2^2}{s_B}} \sinh\left(\frac{2l_1 l_3}{s_A}\right) \quad \text{(S15)}$$

One of the concrete predictions emerging from this calculation is that if $a = 0$, meaning that the pseudoknot is symmetric, that the entropy of the structure should depend on l_2 only as $\exp(-l_2^2/s_B)$ where s_B depends on the lengths of the various loops but is independent of l_1, l_3, and s_5.

We now return to the more general case of $a \neq 0$. In this case, we define a new variable $\vec{y}$ to be the total vector dotted with $\vec{r_{32}}$ in Eq. S12: $\vec{y} = \frac{s_1 s_2 a}{(s_1+s_2)^2}\vec{x} - \frac{a}{s_1+s_2}\vec{r_4}$. Integration over $\vec{r_{32}}$ then yields

$$e^{\Delta S/k_B}(a\neq 0) = \alpha J \pi^3 l_1 l_2 l_3 s_5 e^{-l_1^2(\frac{1}{s_1}+\frac{1}{s_2})-l_2^2\left(\frac{1}{s_1+s_2}+\frac{s_1^2/s_3+s_2^2/s_4}{(s_1+s_2)^2}\right)-l_3^2/s_5} \times$$
$$\int d\vec{r_4}\frac{1}{r_4}e^{-r_4^2(\frac{1}{s_3}+\frac{1}{s_4}+\frac{1}{s_5})}(e^{2l_3 r_4/s_5}-e^{-2l_3 r_4/s_5})\int d\vec{x}\frac{1}{x}e^{-x^2\left[\frac{s_1 s_2}{s_1+s_2}+(\frac{s_1 s_2}{s_1+s_2})^2(\frac{1}{s_3}+\frac{1}{s_4})\right]}(e^{2l_1 x}-e^{-2l_1 x})\times$$
$$e^{2\vec{x}\cdot\vec{r_4}\left(\frac{s_1 s_2}{s_1+s_2}(\frac{1}{s_3}+\frac{1}{s_4})\right)}\frac{1}{y}\left(e^{2l_2 y}-e^{-2l_2 y}\right). \quad \text{(S16)}$$

As before, we can perform three of the angle integrals to yield $8\pi^2$, and define θ to be the angle between $\vec{r_4}$ and $\vec{x}$. Then, $\vec{x}\cdot\vec{r_4}$ becomes $r_4 x\cos\theta$. We can then write y in terms of $\cos\theta$: $y = \sqrt{\vec{y}\cdot\vec{y}} = \frac{a}{s_1+s_2}\sqrt{\frac{s_1^2 s_2^2}{(s_1+s_2)^2}x^2 + r_4^2 - \frac{2s_1 s_2}{s_1+s_2}r_4 x\cos\theta}$. We can thus turn the integration over $\cos\theta$ into an integration over y (again, the Jacobian needs to be accounted for). Defining the limits of the integration to be $y_\pm = \sqrt{\frac{a^2}{(s_1+s_2)^2}\left(\frac{s_1 s_2}{(s_1+s_2)}x \pm r_4\right)^2}$, we have

$$e^{\Delta S/k_B}(a\neq 0)=\alpha J 8\pi^5 l_1 l_2 l_3 s_5\frac{(s_1+s_2)^3}{s_1 s_2 a^2}e^{-l_1^2(\frac{1}{s_1}+\frac{1}{s_2})-l_2^2\left(\frac{1}{s_1+s_2}+\frac{s_1^2/s_3+s_2^2/s_4}{(s_1+s_2)^2}\right)-l_3^2/s_5}\times$$
$$\int_0^\infty dr_4 e^{-r_4^2(\frac{1}{s_3}+\frac{1}{s_4}+\frac{1}{s_5})}(e^{2l_3 r_4/s_5}-e^{-2l_3 r_4/s_5})\int_0^\infty dx e^{-x^2\left(\frac{s_1 s_2 s_{1234}}{(s_1+s_2)^2 s_3 s_4}\right)}(e^{2l_1 x}-e^{-2l_1 x})\times$$
$$\int_{y_-}^{y_+} dy e^{\left(\frac{s_1^2 s_2^2}{(s_1+s_2)^2}x^2+r_4^2-\frac{(s_1+s_2)^2}{a^2}y^2\right)\left(\frac{1}{s_3}+\frac{1}{s_4}\right)}\left(e^{2l_2 y}-e^{-2l_2 y}\right). \quad \text{(S17)}$$

The y integral must be done first because its limits include the other two integration variables. However, this integral results in an error function which cannot be integrated analytically. While various limits might be taken to impose analyticity, given the speeds of programs like Mathematica in performing simple numerical integrals like this one, we prefer to solve the resulting integrals numerically.

There are eight parameters to be varied, and we display the results of the entropy calculation for single-parameter sweeps in Fig. S9. For this figure, we set $s_1=3$, $s_2=4$, $s_3=6$, $s_4=8$, $s_5=3$, $l_1=2$, $l_2=3$, $l_3=4$. Then, keeping all other parameters at those values, we take each parameter and measure the entropy as a function of varying that parameter.

The resulting plot contains eight different curves, which we've plotted in Fig. S9. As expected, for $s_1=s_5=3$, the blue and orange curves coincide. Varying the loop lengths (panel A) appears to give less dramatic changes than varying the stem lengths (panel B). The parameter l_2 was capped at seven because for values greater than that, one of the hairpins wouldn't be able to close ($s_1+s_2<l_2$). The asymmetry between the l_1 and l_3 curves is due to the asymmetry between the constant values of l_1 and l_3 chosen. We also verified that the result of the numerical integration for $a\neq 0$ approaches the result of the analytic solution ($a=0$) as a approaches zero.

We also give a more comprehensive result of the numerical integration. Since displays of eight-parameter tables are difficult to achieve, we give the results of this numerical integration for values of s_i and l_i ranging from 1 to 5 (or s_i' ranging from $1/\gamma$ to $5/\gamma$) as a .h5 file. These types of files can easily be imported using, for example, Python, with the following lines of code:

```
import h5py
import numpy as np
f = h5py.File('kissingHairpinsSuppFile.h5', 'r')
k = np.array(f[list(f.keys())[0]])
```

This code sets the variable `k` to be an eight-dimensional array, such that `k[a][b][c][d][e][f][g][h]` is the entropy (in units of k_B) of a kissing hairpin with $s_1'=(a+1)/\gamma$, $s_2'=(b+1)/\gamma$, ..., $l_1=f+1$, ..., $l_3=h+1$. The addition of 1 is included because Python begins indexing at 0.

We set the two loop entropy parameters to $b=2.4$ and $v_s=0.02$. As mentioned, the entropy is measured in units of Boltzmann's constant k_B.

We also considered the constraints that each hairpin must have ≥ 3 nts (so $s_1+s_2+l_2\geq 4$ and same for s_3 and s_4) and that the hairpins must be able to close (so $s_1+s_2\geq l_2$ and same for s_3 and s_4). We included these constraints by setting the table values to 0 if these constraints aren't satisfied; of course, if these constraints aren't satisfied the entropy should really be considered to be $-\infty$.

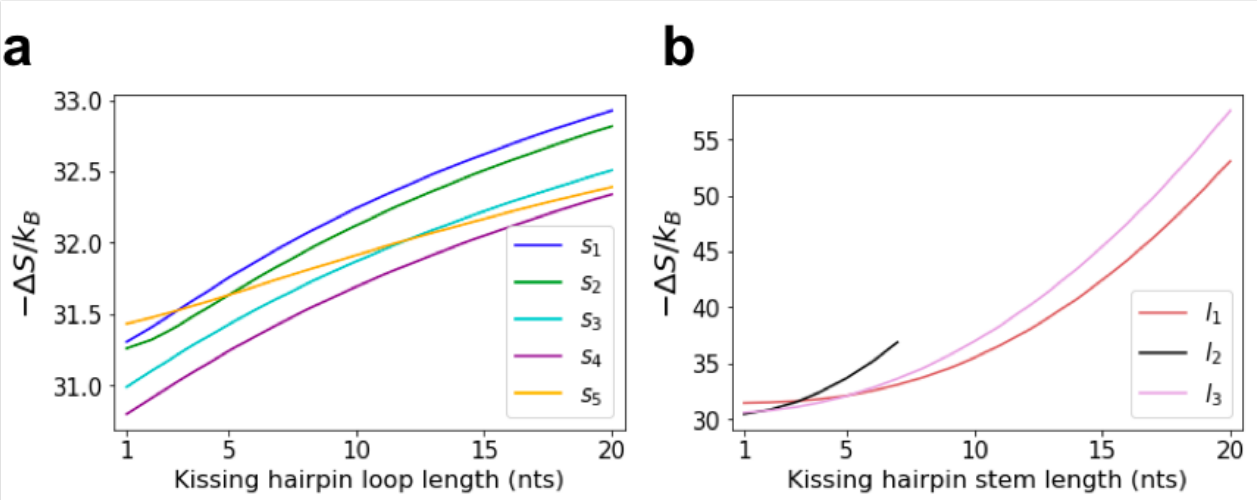

FIG. S9: **One dimensional parameter sweeps for the kissing hairpin pseudoknot entropy.** We set $s_1 = 3$, $s_2 = 4$, $s_3 = 6$, $s_4 = 8$, $s_5 = 3$, $l_1 = 2$, $l_2 = 3$, $l_3 = 4$. Then, keeping all other parameters at those values, we take each parameter and measure the entropy as a function of varying that parameter. See the text for detailed discussion.

S9. CONSIDERING OTHER COMPLEX PSEUDOKNOTS

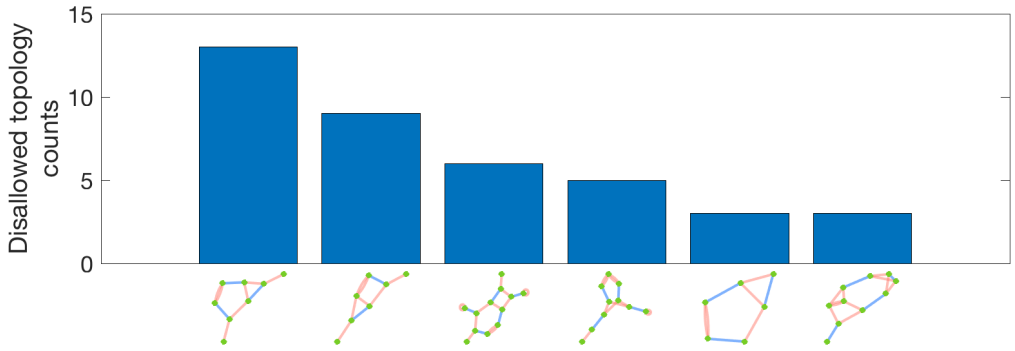

FIG. S10: **Common topologies disallowed by constraints chosen for our algorithm implementation.**

Similar approaches as in the previous section can be taken for other pseudoknots more complex than those shown in Fig. S2.

As discussed, there were 64 pseduoknotted sequences in the experimental datasets used which were found to fold into topologies more complex than those allowed by the constraints we chose to place on the algorithm. Of these 64 sequences, we sought to determine the topologies they shared in common. The six most common topologies and the number of sequences folding into them are plotted in Fig. S10. The most common topology (shown in large in Fig. S8B) is equivalent to an H-type pseudoknot with an internal loop in one stem. As can be seen from Fig. S10, the second and fifth most common topologies are only slight variations on the first: the second is identical to the first with one of the stem lengths set to zero (i.e. the stem is made up of a single base pair) and the fifth is identical to the first with the dangling unpaired regions on the 3' and 5' ends removed.

The entropy of the most common disallowed topology, displayed in large in Fig. S8B, is given by

$$e^{\Delta S/k_B} = v_s^3 \int d\vec{r_1} \int d\vec{r_2} \int d\vec{r_3} \int d\vec{r_4} \int d\vec{r_5} \, \frac{\delta(|\vec{r_1}| - l_1)}{4\pi l_1^2} \frac{\delta(|\vec{r_3} - \vec{r_2}| - l_2)}{4\pi l_2^2} \frac{\delta(|\vec{r_5} - \vec{r_4}| - l_3)}{4\pi l_3^2} \times P_{s_1}(\vec{r_2}) P_{s_2}(\vec{r_2} - \vec{r_1}) P_{s_3}(\vec{r_4} - \vec{r_3}) P_{s_4}(\vec{r_4} - \vec{r_3}) P_{s_5}(\vec{r_5} - \vec{r_1}). \quad \text{(S18)}$$

After changing our integration variables to be $\vec{r_1}$, $\vec{r_{21}}$, $\vec{r_{32}}$, $\vec{r_{45}}$, and $\vec{r_{53}}$, we follow the same formula as for the kissing hairpin pseudoknot to get a similar expression:

$$e^{\Delta S/k_B} = \alpha 8\pi^5 l_1 l_2 l_3 \frac{s_1 s_3 s_4 s_5}{s_3 + s_4} e^{-l_1^2/s_1 - l_2^2/s_5 - l_3^2\left(\frac{1}{s_3}+\frac{1}{s_4}\right)} \int_0^\infty dr_{21} e^{-r_{21}^2\left(\frac{1}{s_1}+\frac{1}{s_2}\right)} \left(e^{2l_1 r_{21}/s_1} - e^{-2l_1 r_{21}/s_1}\right) \times \int_0^\infty dr_{53} e^{-r_{53}^2\left(\frac{1}{s_3}+\frac{1}{s_4}\right)} \left(e^{2\left(\frac{1}{s_3}+\frac{1}{s_4}\right) l_3 r_{53}} - e^{-2\left(\frac{1}{s_3}+\frac{1}{s_4}\right) l_3 r_{53}}\right) \int_{y_-}^{y_+} dy e^{-y^2/s_5} \left(e^{2l_2 y/s_5} - e^{-2l_2 y/s_5}\right) \quad \text{(S19)}$$

where $y_\pm = \sqrt{(r_{53} \pm r_{21})^2}$.

Using this formula, we find that if one instead considers the entropy of the H-type pseudoknot with an internal loop to be comprised of the sum of the entropies of the H-type pseudoknot and the internal loop, this leads to an overestimate of the entropy cost of at least 1 kcal/mol over nearly all parameter values at 37°C. This overestimate is significantly higher for some parameters; a representative example is the case of $l_1 = 2$, $l2 = 4$, $l_3 = 4$, $s_1 = 3$, $s_2 = 3$, (the results are fairly insensitive to s_3, s_4, s_5) which yields an entropy difference of 3.3 kcal/mol, or a 23% error. Changing these parameters can both increase or decrease this error, but there is a very wide parameter regime in which the error due to not taking into account the nestedness of the internal loop is significant.

S10. SAMPLE FREE ENERGY CALCULATION AND GRAPH DECOMPOSITION PROCESS

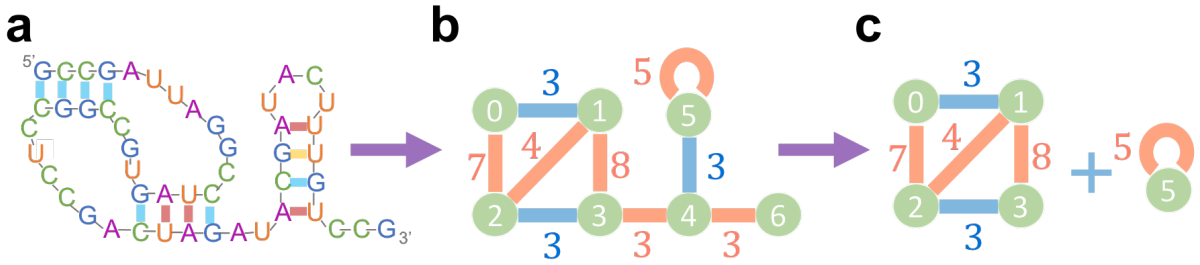

FIG. S11: **Sample structure.** **A** structure under consideration; **B** graph representing the structure; **C** fully decomposed graph. The loop entropy of the structure is the sum of the loop entropies of the graphs in this panel.

Here we describe the graph decomposition process – the basis for the loop entropy calculation in practice – in some more detail and provide a sample calculation of the free energy as an example.

Given a structure (graph) we test each possible edge for whether removing that edge leads to a disconnected graph. If so, we remove it, and the two resulting graphs represent two different motifs. We repeat, and compare the final graphs (that cannot be decomposed further) to our tabulated list (Fig. S2); some of these graphs may represent pseudoknots, while others represent hairpins. Thus, using our tabulated or analytically calculated results for the loop entropy of each possible graph, we calculate the loop entropy of each motif in the RNA structure, and sum them to find the total loop entropy.

As an example, let's consider the structure shown in Fig. S11A. We'd like to calculate the free energy of this structure. First, we calculate the enthalpy terms using the Turner parameters. These include a dangling end, as well as stacking terms and terminal mismatches:

G . G C . C C . C G . G A
C C ' C G ' G G ' G C ' C C

C C . C U . U A . A G . G A
A G ' G A ' A U ' U C ' C **C**

U A . A C . C G . G A . A U
C U ' U G ' G U ' U U ' U U

where the top line goes from 5' to 3' and the bottom line is antiparallel. The bolded C (last in second row) represents the approximation in the Turner rules that if two base pairs can bind but are unbound in the structure, the purine is replaced with A and the pyrimidine with C. Each of these terms has an associated enthalpy and entropy from the tabulated Turner parameters.

Once these terms have been added up, the remaining step is calculating the free energy of the loops. First, we convert the structure to a graph (Fig. S11B) by placing nodes at the edges of stems (here we also place nodes at the ends of the sequence). These nodes are connected by double-stranded (blue) or single-stranded (red) edges. In fact, since the stems have at least length 1, each node (except for perhaps the ones representing the edges of the molecule) must be connected to one double-stranded and two single-stranded edges; the hairpin loop counts as two edges for this purpose. The lengths of the various edges are provided in the figure.

Now, we perform the graph decomposition process. We test each possible edge for whether removing that edge leads to a disconnected graph. The first edge for which this is true is that connecting nodes three and four. We therefore remove that edge, and the two resulting graphs represent two different motifs. We repeat, finding that removing the edge between nodes four and five similarly disconnects the graphs, and same for the edge between nodes four and six. Finally, finding that nodes four and six are not connected to any edges we remove those. We compare the final graphs that cannot be decomposed further – Fig. S11C – to our tabulated list (Fig. S2). We find here that we have one instance of an open-net-2a ($l_1 = 3$; $l_2 = 3$; $s_1 = 7$; $s_2 = 4$; $s_3 = 8$) and a closed-net-0 ($s_1 = 5$). This gives us the loop entropy resulting from this structure, which we add to the bond entropy found using the Turner parameters to

get the total free energy of the structure.

---

[1] Mathai Mammen, Eugene I. Shakhnovich, John M. Deutch, and George M. Whitesides. Estimating the Entropic Cost of Self-Assembly of Multiparticle Hydrogen-Bonded Aggregates Based on the Cyanuric Acid-Melamine Lattice. *Journal of Organic Chemistry*, 63(12):3821–3830, 1998.

[2] Huan-xiang Zhou and Michael K Gilson. Theory of Free Energy and Entropy in Noncovalent Binding. *Chemical Reviews*, 109(9):4092–4107, 2009.

[3] Michael Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–3415, 2003.

[4] John SantaLucia and Donald Hicks. The Thermodynamics of DNA Structural Motifs. *Annual Review of Biophysics and Biomolecular Structure*, 33(1):415–440, 2004.

[5] Naoki Sugimoto, Shu ichi Nakano, Misa Katoh, Akiko Matsumura, Hiroyuki Nakamuta, Tatsuo Ohmichi, Mari Yoneyama, and Muneo Sasaki. Thermodynamic Parameters To Predict Stability of RNA/DNA Hybrid Duplexes. *Biochemistry*, 34(35):11211–11216, 1995.

[6] Norman E. Watkins, William J. Kennelly, Mike J. Tsay, Astrid Tuin, Lara Swenson, Hyung Ran Lee, Svetlana Morosyuk, Donald A. Hicks, and John SantaLucia. Thermodynamic contributions of single internal rA·dA, rC·dC, rG·dG and rU·dT mismatches in RNA/DNA duplexes. *Nucleic Acids Research*, 39(5):1894–1902, 2011.

[7] Homer Jacobson and Walter H. Stockmayer. Intramolecular reaction in polycondensations. I. The theory of linear systems. *The Journal of Chemical Physics*, 18(12):1600–1606, 1950.

[8] H. Isambert and E. D. Siggia. Modeling RNA folding paths with pseudoknots: Application to hepatitis delta virus ribozyme. *Proceedings of the National Academy of Sciences*, 97(12):6515–6520, 2000.

[9] A. Xayaphoummine, T. Bucher, and Herve Isambert. Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots. *Nucleic Acids Research*, 33(SUPPL. 2):605–610, 2005.