Scalable and Interpretable Identification of Minimal Undesignable RNA Structure Motifs with Rotational Invariance

Tianshuo Zhou[1], Wei Yu Tang[1,6], Apoorv Malik[1], David H. Mathews[3,4,5], and Liang Huang[1,2]

[1] School of EECS and [2] Dept. of Biochemistry & Biophysics, Oregon State University, Corvallis OR 97330, USA
[3] Dept. of Biochemistry & Biophysics, [4] Center for RNA Biology, and [5] Dept. of Biostatistics & Computational Biology, University of Rochester Medical Center, Rochester, NY 14642, USA
[6] Department of Quantitative and Computational Biology, University of Southern California, CA 90089, USA

Abstract. RNA design aims to find a sequence that folds with highest probability into a designated target structure. However, certain structures are *undesignable*, meaning no sequence can fold into the target structure under the default (Turner) RNA folding model. Understanding the specific local structures (i.e., "motifs") that contribute to undesignability is crucial for refining RNA folding models and determining the limits of RNA designability. Despite its importance, this problem has received very little attention, and previous efforts are neither scalable nor interpretable.

We develop a new theoretical framework for motif (un-)designability, and design scalable and interpretable algorithms to identify minimal undesignable motifs within a given RNA secondary structure. Our approach establishes motif undesignability by searching for rival motifs, rather than exhaustively enumerating all (partial) sequences that could potentially fold into the motif. Furthermore, we exploit rotational invariance in RNA structures to detect, group and reuse equivalent motifs and to construct a database of unique minimal undesignable motifs. To achieve that, we propose a loop-pair graph representation for motifs and a recursive graph isomorphism algorithm for motif equivalence.

Our algorithms successfully identify 24 unique minimal undesignable motifs among 18 undesignable puzzles from the Eterna100 benchmark. Surprisingly, we also find over 350 unique minimal undesignable motifs and 663 undesignable native structures in the ArchiveII dataset, drawn from a diverse set of RNA families.

Availability: Our source code is available at `https://github.com/shanry/RNA-Undesign` and our server is available at `http://linearfold.org/motifs`.
Appendix: Supplementary material is available in a separate file.

Keywords: RNA Design · Inverse Folding · Undesignability · Designability · Structure Motif.

1 Introduction

RNA design [34,4,22,12,32] focuses on identifying one or more RNA sequences capable of folding into a target secondary structure, holding significant promise for applications in RNA-based therapeutics and diagnostics [33].

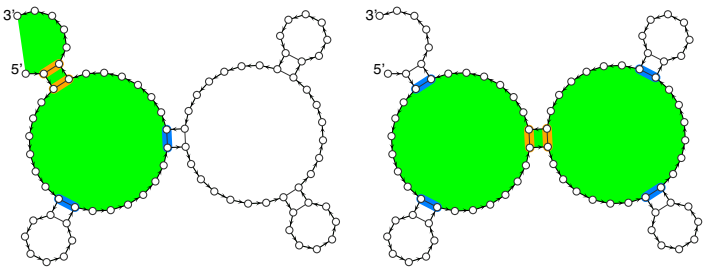


Fig. 1: Going beyond structure undesignability in our previous work [35], this work finds *minimal undesignable motifs* in a given RNA structure (two such motifs shown here in Eterna100 puzzle #52). Boundary and internal pairs are in orange and blue, resp.

The significance and complexity of RNA design have garnered widespread attention. Numerous methods, including SAMFEO [34], NEMO [22], RNAiFold [12], NUPACK [32], and others [4], have been developed to generate sequences based on a given target structure. Despite substantial improvements in empirical design quality, it has been noted that certain puzzles in the widely-used benchmark Eterna100 [2] are considered undesignable, having never been successfully solved [17]. However, limited research [1] has been dedicated to exploring the undesignability of RNA structures. Recently, We introduced RIGEND [35] to identify undesignable RNA structures in a more general manner by pinpointing rival structures, resulting in the identification of 16 puzzles in Eterna100 deemed undesignable by the unique minimum free energy (MFE) criterion under the standard Turner RNA folding model [21,26] implemented in ViennaRNA [20] .

While effective, RIGEND has some limitations in terms of explainability and scalability. While rival structures, which consistently exhibit better folding energy compared to the target structure, can serve as compelling evidence for an undesignable structure, their interpretability is often limited to the entire structure. In practice, the undesignability of an RNA structure typically arises from some specific local region, or *structure motif*. Identifying these critical motif(s) in a structure could offer deeper insight into why certain structures resist design, enhancing both interpretability and potential reusability in RNA Design. However, important as it is, this problem has received very little attention, and the existing efforts for motif designability [31,30] are neither scalable nor interpretable (see Sec. 7 for details).

To address these limitations, we conduct a systematic study of undesignable motifs, introducing general theories and efficient algorithms for identifying *minimal undesignable motifs* (examples shown in Fig. 1) from given RNA secondary structures. Our contributions are as follows:

1. We develop a new theoretical framework for RNA motif (un-)designability, including new definitions and theorems.
2. We propose fast algorithms to identify motifs that are undesignable by searching for rival motifs, offering strong explainability.
3. We exploit rotational invariance in RNA structures to detect, group, and reuse equivalent motifs. To achieve that, we introduce a loop-pair graph representation for motifs and develop a recursive graph isomorphism algorithm to identify unique (undesignable) motifs.
4. We develop an efficient bottom-up scan algorithm called FastMotif to identify minimal undesignable motifs from RNA structures, with an average time cost of less than 10 seconds per structure in experiments.
5. We identify 24 unique minimal undesignable motifs from 18 puzzles in the Eterna100 benchmark. Moreover, we identify 331 unique minimal undesignable motifs in natural RNA structures from ArchiveII, revealing limitations in the Turner nearest neighbor energy model. In total there are 355 unique minimal undesignable motifs with explainable rivals motifs, and the majority (300+) of them were never proven undesignable before this work.

2 RNA Structure and its Undesignability

2.1 Secondary Structure, Loop and Free Energy

An RNA sequence $\boldsymbol{x}$ of length n is specified as a string of base nucleotides $\boldsymbol{x}_1\boldsymbol{x}_2\ldots\boldsymbol{x}_n$, where $\boldsymbol{x}_i \in \{\mathrm{A},\mathrm{C},\mathrm{G},\mathrm{U}\}$ for $i = 1, 2, ..., n$. A secondary structure $\mathcal{P}$ for $\boldsymbol{x}$ is a set of paired indices where each pair $(i, j) \in$

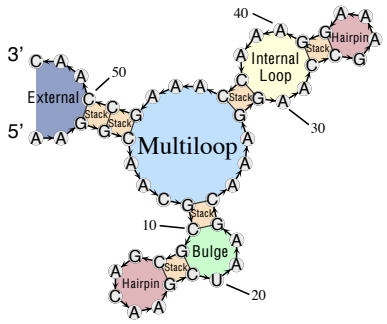


Fig. 2: RNA secondary structure and loops.

Table 1: Critical positions of some loops in Fig. 2.

Loop	Critical Positions
Hairpin$\langle(12, 18)\rangle$	(12, 18), 13, 17
Bulge$\langle(10, 23), (11, 19)\rangle$	(10, 23), (11, 19)
Stack$\langle(3, 50), (4, 49)\rangle$	(3, 50), (4, 49)
Internal$\langle(29, 43), (32, 39)\rangle$	(29, 43), (32, 39), 30, 42, 31, 40
External$\langle(3, 50)\rangle$	(3, 50), 2, 51
...	...

$\mathcal{P}$ indicates two distinct bases $\boldsymbol{x}_i\boldsymbol{x}_j \in \{\text{CG}, \text{GC}, \text{AU}, \text{UA}, \text{GU}, \text{UG}\}$ and each index from 1 to n can only be paired once. A secondary structure is pseudoknot-free if there are no two pairs $(i, j) \in \mathcal{P}$ and $(k, l) \in \mathcal{P}$ such that $i < k < j < l$. Alternatively, $\mathcal{P}$ can be represented as a string $\boldsymbol{y} = \boldsymbol{y}_1\boldsymbol{y}_2 \dots \boldsymbol{y}_n$, where a pair of indices $(i, j) \in \mathcal{P}$ corresponds to $\boldsymbol{y}_i =$ "(", $\boldsymbol{y}_j =$ ")" and any unpaired index k corresponds to $\boldsymbol{y}_k =$ ".". The unpaired indices in $\boldsymbol{y}$ are denoted as $\mathit{unpaired}(\boldsymbol{y})$ and the set of paired indices in $\boldsymbol{y}$ is denoted as $\mathit{pairs}(\boldsymbol{y})$, which is equal to $\mathcal{P}$. While some RNA structures in nature contain pseudoknots, we do not consider them in this work as the computational model we use does not allow these.

The *ensemble* of an RNA sequence $\boldsymbol{x}$ is the set of all secondary structures that $\boldsymbol{x}$ can possibly fold into, denoted as $\mathcal{Y}(\boldsymbol{x})$. The *free energy* $\Delta G^\circ(\boldsymbol{x}, \boldsymbol{y})$ is used to characterize the stability of $\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})$. The lower the free energy $\Delta G^\circ(\boldsymbol{x}, \boldsymbol{y})$, the more stable the secondary structure $\boldsymbol{y}$ for $\boldsymbol{x}$. In the nearest neighbor energy model [26], a secondary structure is decomposed into a collection of loops, where each loop is usually a region enclosed by some base pair(s). Depending on the number of pairs on the boundary, the main types of loops include hairpin loop, internal loop and multiloop, which are bounded by 1, 2 and 3 or more base pairs, respectively. In particular, the external loop is the most outside loop and is bounded by two ends ($5'$ and $3'$) and other base pair(s). Thus each loop can be identified by a set of pairs. Fig. 2 showcases an example of secondary structure with loops, and some of the loops are listed in Table 1.

The function $\mathit{loops}(\boldsymbol{y})$ is used to denote the set of loops in a structure $\boldsymbol{y}$. The free energy of a secondary structure $\boldsymbol{y}$ is the sum of the free energy of each loop, $\Delta G^\circ(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{z} \in \mathit{loops}(\boldsymbol{y})} \Delta G^\circ(\boldsymbol{x}, \boldsymbol{z})$, where each term $\Delta G^\circ(\boldsymbol{x}, \boldsymbol{z})$ is the energy for one specific loop in $\mathit{loops}(\boldsymbol{y})$. See Supplementary Section G for detailed energy functions for types of loops in the Turner model [26]. The energy of each loop is typically determined by the identity of nucleotides on the positions of enclosing pairs and their adjacent mismatch positions, which are named as *critical positions* in this article. Table SI 1 lists the critical positions for all the loops in Fig. 2 and Table SI 2 shows the indices of critical positions for each type of loops. Additionally, some *special hairpins* [21] of unstable triloops and stable tetraloops and hexaloops in Turner model have a separate energy lookup table (See Supplementary Section G.2). When evaluating the energy of a loop, it suffices to input only the nucleotides on its critical positions, i.e.,

$$\Delta G^\circ(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{z} \in \mathit{loops}(\boldsymbol{y})} \Delta G^\circ(\boldsymbol{x}, \boldsymbol{z}) = \sum_{\boldsymbol{z} \in \mathit{loops}(\boldsymbol{y})} \Delta G^\circ(\boldsymbol{x} \vdash \mathit{critical}(\boldsymbol{z}), \boldsymbol{z}), \tag{1}$$

where $\mathit{critical}(\boldsymbol{z})$ denotes the critical positions of loop $\boldsymbol{z}$ and $\boldsymbol{x} \vdash \mathit{critical}(\boldsymbol{z})$ denotes the nucleotides from $\boldsymbol{x}$ that are "projected" onto $\mathit{critical}(\boldsymbol{z})$. See Supplementary Section B for the detailed functionality of projection operator. The projection ($\vdash$) allows us to focus on the relevant nucleotides for energy evaluation. Table 1 list the critical positions for the example loops.

2.2 Undesignability by the Unique MFE Criterion

The structure with the *minimum free energy* is the most stable structure in the ensemble. A structure $\boldsymbol{y}^\star$ is an MFE structure of $\boldsymbol{x}$ if and only if

$$\forall \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}) \text{ and } \boldsymbol{y} \neq \boldsymbol{y}^\star, \Delta G^\circ(\boldsymbol{x}, \boldsymbol{y}^\star) \leq \Delta G^\circ(\boldsymbol{x}, \boldsymbol{y}). \tag{2}$$

RNA design is the inverse problem of RNA folding. Given a target structure $\boldsymbol{y}^\star$, RNA design aims to find suitable RNA sequence $\boldsymbol{x}$ such that $\boldsymbol{y}^\star$ is an MFE structure of $\boldsymbol{x}$. Here we follow a more strict definition of MFE criterion adopted in some previous studies [6,14,31,27,34] on the designability of RNA, i.e., $\boldsymbol{x}$ is a correct design if and only if $\boldsymbol{y}$ is the only MFE structure of $\boldsymbol{x}$, which we call unique MFE (uMFE) criterion to differentiate it from the traditional MFE criterion. Formally, $\text{uMFE}(x) = \boldsymbol{y}^\star$ if and only if

$$\forall \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}) \text{ and } \boldsymbol{y} \neq \boldsymbol{y}^\star, \Delta G^\circ(\boldsymbol{x}, \boldsymbol{y}^\star) < \Delta G^\circ(\boldsymbol{x}, \boldsymbol{y}). \tag{3}$$

Following previous work [14,35] on undesignability, we define the undesignability based the uMFE criterion, i.e., the condition in Eq.3 can not be satisfied for any RNA sequence $\boldsymbol{x}$ given a target structure $\boldsymbol{y}^\star$. Alternatively, we give the formal definition of undesignability as follows.

Definition 1. *An RNA secondary structure $\boldsymbol{y}^\star$ is undesignable by* uMFE *criterion if and only if*

$$\forall \boldsymbol{x}, \exists \boldsymbol{y'} \neq \boldsymbol{y}^\star, \Delta G^\circ(\boldsymbol{x}, \boldsymbol{y'}) \leq \Delta G^\circ(\boldsymbol{x}, \boldsymbol{y}^\star). \tag{4}$$

3 Motif and its Undesignability

Recent work, RIGEND [35], has demonstrated that some structures (puzzles) in Eterna100 are undesignable. For instance, the puzzle "`[RNA] Repetitive Seqs. 8/10`" in Fig. 1 is proven undesignable because there exists a rival structures always better a lower free energy than the target structure . However, this level of explainability remains at the global structure level rather than more specific local regions. Ideally, a solution should not only verify that a structure is undesignable but also pinpoint local regions as small as possible within structures that causes the undesignability. We refer to these regions as *undesignable motifs*.

Smaller undesignable motifs are easier to be resused to detect undesignability in structures. Thus, the goal is to identify minimal undesignable motifs from structures. For example, our proposed algorithm identified two minimal undesignable motifs in the puzzle "`[RNA] Repetitive Seqs. 8/10`", highlighted in Fig. 1. Previous efforts [31,30] of exhaustive search fails at them due to scalability and deficit of motif definition.

We argue that the most effective way to define a motif is by focusing on the composition of loops, aligning with how RNA structures are composed of different loop types (Section 2). In this way, a motif is regarded a full generalization of RNA structure. Our formal definition of motifs is introduced in the following subsection.

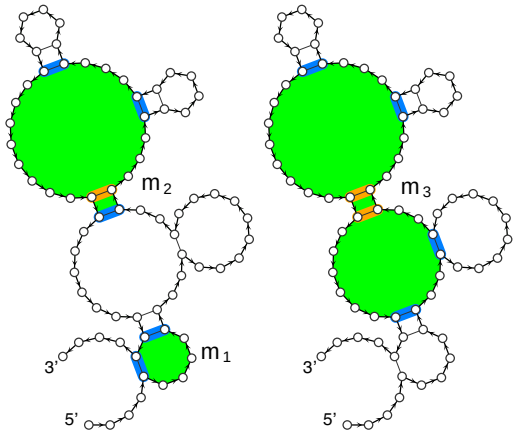


Fig. 3: Motifs with various cardinalities (numbers of loops): $card(\boldsymbol{m}_1)=1$, $card(\boldsymbol{m}_2)=2$, $card(\boldsymbol{m}_3)=3$. Loops are in green, internal pairs (*ipairs*) in orange and boundary pairs (*bpairs*) in blue.

3.1 Motif is a Generalization of Structure

Definition 2. *A motif $\boldsymbol{m}$ is a contiguous (sub)set of loops in an RNA secondary structure $\boldsymbol{y}$, notated $\boldsymbol{m} \subseteq \boldsymbol{y}$.*

Many functions defined for secondary structures can also be applied to motifs. For example, $loops(\boldsymbol{m})$ represents the set of loops within a motif $\boldsymbol{m}$, while $pairs(\boldsymbol{m})$ and $unpaired(\boldsymbol{m})$ represent the sets of base pairs and unpaired positions, respectively. We define the *cardinality* of $\boldsymbol{m}$ as the number of loops in $\boldsymbol{m}$, i.e., $card(\boldsymbol{m}) = |loops(\boldsymbol{m})|$. Fig. 3 illustrates three motifs, $\boldsymbol{m}_1, \boldsymbol{m}_2$, and $\boldsymbol{m}_3$, in a structure adapted from the Eterna puzzle "`Cat's Toy`". These motifs contain 1, 2, and 3 loops, respectively. We also define the *length* of a motif $|\boldsymbol{m}|$ as the number of bases it contains, which is consistent with the length of a secondary structure $|\boldsymbol{y}|$.

Since motifs are defined as sets of loops, we can conveniently use set relations to describe their interactions. A motif $\boldsymbol{m}_A$ is a *sub-motif* of another motif $\boldsymbol{m}_B$ if $\boldsymbol{m}_A$ is contained within $\boldsymbol{m}_B$, denoted as $\boldsymbol{m}_A \subseteq \boldsymbol{m}_B$. For the motifs in Fig. 3, we observe the relation $\boldsymbol{m}_2 \subseteq \boldsymbol{m}_3$. We further use $\boldsymbol{m}_A \subset \boldsymbol{m}_B$ to indicate that $\boldsymbol{m}_A$ is a proper sub-motif of $\boldsymbol{m}_B$, meaning $\boldsymbol{m}_A \neq \boldsymbol{m}_B$. Therefore, $\boldsymbol{m}_2 \subset \boldsymbol{m}_3$. The entire structure $\boldsymbol{y}$ can be regarded as the largest motif within itself, and accordingly, $\boldsymbol{m} \subseteq \boldsymbol{y}$ signifies that motif $\boldsymbol{m}$ is a part of structure $\boldsymbol{y}$, with $\boldsymbol{m} \subset \boldsymbol{y}$ implying $\boldsymbol{m}$ is strictly smaller than $\boldsymbol{y}$.

The loops in a motif $\boldsymbol{m}$ are connected by base pairs. Each base pair in $pairs(\boldsymbol{m})$ is classified as either an *internal pair* linking two loops in $\boldsymbol{m}$ or a *boundary pair* connecting one loop inside $\boldsymbol{m}$ to one outside. These two types of pairs in $\boldsymbol{m}$ are denoted as disjoint sets $ipairs(\boldsymbol{m})$ and $bpairs(\boldsymbol{m})$, respectively:

$$ipairs(\boldsymbol{m}) \cap bpairs(\boldsymbol{m}) = \varnothing, \;\; ipairs(\boldsymbol{m}) \cup bpairs(\boldsymbol{m}) = pairs(\boldsymbol{m}). \tag{5}$$

Utilizing the commonly accepted nearest neighbor model for RNA folding, it becomes evident that certain motifs may be absent from structures folded from RNA sequences. For instance, motif $\boldsymbol{m}_3$ in Fig. 3 is considered undesignable, as the removal of its two internal pairs consistently reduces the free energy. This brings us to the definition of an *undesignable motif*.

3.2 Motif Ensemble from Constrained Folding

The designability of motifs is based on *constrained folding*. Given a sequence $\boldsymbol{x}$, a structure in its ensemble $\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})$, we can conduct constrained folding by constraining the loops outside $\boldsymbol{m}$, i.e., $\boldsymbol{c} = \boldsymbol{y} \setminus \boldsymbol{m}$. We generalize the concept of (structure) *ensemble* to *motif ensemble* as the set of motifs that $\boldsymbol{x}$ can possibly fold into given the constraint $\boldsymbol{c}$, denoted as $\mathcal{M}(\boldsymbol{x}, \boldsymbol{c})$. Motifs in $\mathcal{M}(\boldsymbol{x}, \boldsymbol{y} \setminus \boldsymbol{m})$ have the same boundary pairs, i.e.,

$$\forall \boldsymbol{m}', \boldsymbol{m}'' \in \mathcal{M}(\boldsymbol{x}, \boldsymbol{y} \setminus \boldsymbol{m}), bpairs(\boldsymbol{m}') = bpairs(\boldsymbol{m}'') = bpairs(\boldsymbol{m}). \tag{6}$$

The *free energy* of a motif $\boldsymbol{m}$ is the sum of the free energy of the loops in $\boldsymbol{m}$,

$$\Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}) = \sum_{\boldsymbol{z} \in loops(\boldsymbol{m})} \Delta G^\circ(\boldsymbol{x}, \boldsymbol{z}). \tag{7}$$

The definitions of MFE and uMFE can also be generalized to motifs via constrained folding.

Definition 3. *A motif $\boldsymbol{m}^\star \subseteq \boldsymbol{y}$ is an* MFE *motif of folding $\boldsymbol{x}$ under constraint $\boldsymbol{y} \setminus \boldsymbol{m}^\star$, i.e.,* $\text{MFE}(\boldsymbol{x}, \boldsymbol{y} \setminus \boldsymbol{m}^\star)$*, if and only if*

$$\forall \boldsymbol{m} \in \mathcal{M}(\boldsymbol{x}, \boldsymbol{y} \setminus \boldsymbol{m}^\star) \text{ and } \boldsymbol{m} \neq \boldsymbol{m}^\star, \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}^\star) \leq \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}). \tag{8}$$

Definition 4. *A motif $\boldsymbol{m}^\star \subseteq \boldsymbol{y}$ is an* uMFE *motif of folding $\boldsymbol{x}$ under constraint $\boldsymbol{y} \setminus \boldsymbol{m}^\star$, i.e.,* $\text{uMFE}(\boldsymbol{x}, \boldsymbol{y} \setminus \boldsymbol{m}^\star)$*, if and only if*

$$\forall \boldsymbol{m} \in \mathcal{M}(\boldsymbol{x}, \boldsymbol{y} \setminus \boldsymbol{m}^\star) \text{ and } \boldsymbol{m} \neq \boldsymbol{m}^\star, \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}^\star) < \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}). \tag{9}$$

3.3 Undesignability of Motif

The designability and undesignability of motifs by uMFE criterion can be defined based on Def. 4.

Definition 5. *A motif $\boldsymbol{m}^\star \subseteq \boldsymbol{y}$ is an undesignable motif by* uMFE *criterion if and only if*

$$\forall \boldsymbol{x}, \exists \boldsymbol{m}' \neq \boldsymbol{m}^\star, \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}') \leq \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}^\star). \tag{10}$$

Similarly, we can establish the definition of designable motifs.

Definition 6. *A motif $\boldsymbol{m}^\star \subseteq \boldsymbol{y}$ is a designable motif by* uMFE *criterion if and only if*

$$\exists \boldsymbol{x}, \boldsymbol{m}^\star = \text{uMFE}(\boldsymbol{x}, \boldsymbol{y} \setminus \boldsymbol{m}^\star). \tag{11}$$

Moreover, if $\boldsymbol{m}$ is undesignable, any motif or structure containing $\boldsymbol{m}$ will be undesignable.

Theorem 1. *If a motif $\boldsymbol{m}^\star$ is undesignable, then any motif $\boldsymbol{m}$ such that $\boldsymbol{m}^\star \subseteq \boldsymbol{m}$ is undesignable.*

Proof. By Def. 5, $\forall \boldsymbol{x}, \exists \boldsymbol{m}' \neq \boldsymbol{m}^\star, \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}') \leq \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}^\star)$. We can construct a motif $\boldsymbol{m}'' \neq \boldsymbol{m}$ by substituting the loops of $\boldsymbol{m}^\star$ within $\boldsymbol{m}$ with $\boldsymbol{m}''$ such that $loops(\boldsymbol{m}'') = loops(\boldsymbol{m}) \setminus loops(\boldsymbol{m}^\star) \cup loops(\boldsymbol{m}')$. As a result, $\forall \boldsymbol{x}, \exists \boldsymbol{m}'' \neq \boldsymbol{m}, \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}'') - \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}) = \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}') - \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}^\star) \leq 0$. □

Corollary 1. *If a motif $\boldsymbol{m}^\star$ is undesignable, then any structure $\boldsymbol{y}$ such that $\boldsymbol{m}^\star \subseteq \boldsymbol{y}$ is also undesignable.*

Proof. The structure $\boldsymbol{y}$ is the largest motif in $\boldsymbol{y}$. Thus the correctness of Corollary 1 follows Theorem 1. □

Corollary 2. *If a motif $\boldsymbol{m}$ is designable, then any sub-motif $\boldsymbol{m}_{sub} \subset \boldsymbol{m}$ is also designable. As a special case, if a structure $\boldsymbol{y}$ is designable, then any motif $\boldsymbol{m}$ within $\boldsymbol{y}$, i.e., $\boldsymbol{m} \subset \boldsymbol{y}$ is also designable.*

Proof. This follows as the contrapositive of Theorem 1, thus completing the proof. □

According to Theorem 1, we can access the undesignability of a big motif by inspecting its sub-motifs. Therefore, it is crucial and valuable to determine the minimality of an undesignable motif. We introduce the concept of *minimal undesignable motif*.

Definition 7. *A motif $\boldsymbol{m}$ is a minimal undesignable motif if and only if the two conditions both hold: (1) $\boldsymbol{m}$ is an undesignable motif, and (2) $\forall \boldsymbol{m}' \subset \boldsymbol{m}$, $\boldsymbol{m}'$ is designable.*

By this definition, the motif $\boldsymbol{m}_3$ in Fig. 3 is a minimal undesignable motif because all its proper sub-motifs are designable. Since the minimality is based on the concept of loops, one fundamental question is that what's the least number of loops a minimal undesignable motif can contain. Therefore, it is worthwhile to prove that any motif composed of a single loop is designable.

Theorem 2. *If a motif $\boldsymbol{m}^\star$ is composed of one loop, i.e., $card(\boldsymbol{m}^\star) = 1$, then $\boldsymbol{m}^\star$ is designable.*

Proof is detailed in Supplementary Section F.2. It turns out that it is possible for a two-loop motif to be minimal undesignable (as illustrated in Table 3). The primary issue is how to identify undesignable motifs. A trivial solution is to enumerate all the (partial) sequences for the target motif $\boldsymbol{m}^\star$ and check the when $\boldsymbol{m}^\star$ is a uMFE motif after constrained folding. However, it is impractical for long motifs because of exponentially high time cost. See Supplementary Section D for a detailed discussion. The existing work CountingDesign [31,30] is better than exhaustively enumerating sequences for each motif one by one, yet in essence it is still an exhaustive enumeration method. As a result, it cannot scale to long motifs and the found undesignable motifs lack interpretability. To provide scalability but also interpretability, we borrow the philosophy of rival structure from RIGEND [35] and propose to utilize rival motif to establish the undesignability of motifs, which is discussed in the following Section 4.

4 Rival Motifs Identification

4.1 Identify Single Rival Motif

It is possible that there is another motif $\boldsymbol{m}'$ which always has lower free energy than the target motif $\boldsymbol{m}^\star$, if we can find such a *rival motif*, $\boldsymbol{m}^\star$ is undesignable. For instance, removing the internal pairs of the motif $\boldsymbol{m}_3$ highlighted in Fig. 3 will yield a rival motif $\boldsymbol{m}'$ that is always energetically favored than $\boldsymbol{m}_3$, proving $\boldsymbol{m}_3$ in Fig. 3 undesignable by the following theorem. Another example of single rival motif is shown in Fig. 4b.

Theorem 3. *If* $\exists \boldsymbol{m}' \neq \boldsymbol{m}^\star, \forall \boldsymbol{x}, \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}') \leq \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}^\star)$*, then* $\boldsymbol{m}^\star$ *is undesignable.*

The correctness of Theorem 3 follows Def. 5. According to RIGEND [35], the energy difference of two structures $\Delta\Delta G^\circ$ is totally decided by their *differential positions*, which can also be applied to two motifs. The condition in Theorem 3 can be written as

$$\exists \boldsymbol{m}' \neq \boldsymbol{m}^\star, \forall \boldsymbol{x}' = \boldsymbol{x} \vdash \Delta(\boldsymbol{m}', \boldsymbol{m}^\star), \Delta\Delta G^\circ(\boldsymbol{x}', \boldsymbol{m}', \boldsymbol{m}^\star) \leq 0 \ , \tag{12}$$

where

$$\Delta(\boldsymbol{m}', \boldsymbol{m}^\star) \triangleq \bigcup_{z \in loops(\boldsymbol{m}^\star) \ominus loops(\boldsymbol{m}')} critical(z) \tag{13}$$

denotes the *differential positions*[1], and

$$\Delta\Delta G^\circ(\boldsymbol{x}', \boldsymbol{m}', \boldsymbol{m}^\star) \triangleq \sum_{z' \in loops(\boldsymbol{m}') \setminus loops(\boldsymbol{m}^\star)} \Delta G^\circ(\boldsymbol{x}' \vdash critical(z'), z') - \sum_{z^\star \in loops(\boldsymbol{m}^\star) \setminus loops(\boldsymbol{m}')} \Delta G^\circ(\boldsymbol{x}' \vdash critical(z^\star), z^\star) \tag{14}$$

denotes the *free energy difference* between $\boldsymbol{m}$ and $\boldsymbol{m}^\star$. Therefore, we can verify a single rival motif by inspecting possible nucleotides on only these differential positions. Refer to the RIGEND [35] for details.

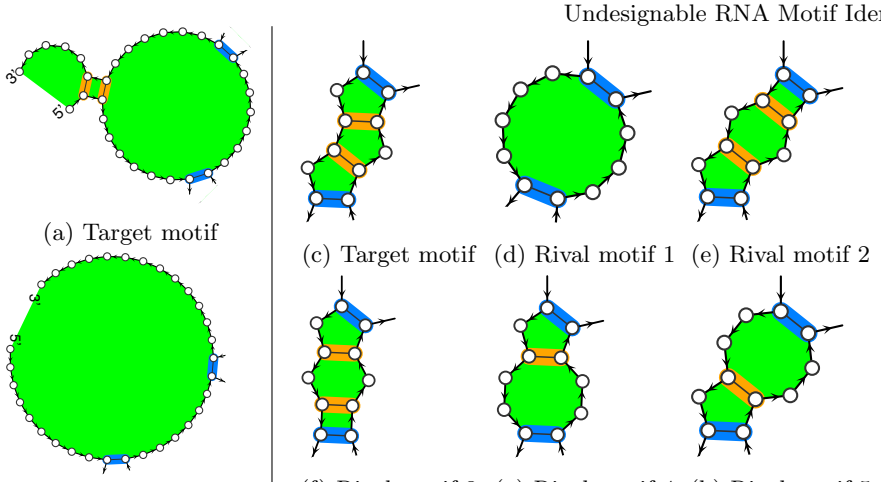
4.2 Identify Multiple Rival Motifs

For a rival motif $\boldsymbol{m}'$ satisfying Theorem 3, we have $pairs(\boldsymbol{m}') \subset pairs(\boldsymbol{m}^\star)$ (proven in Supplementary Section E). However, if any motif $\boldsymbol{m}'$ satisfying $pairs(\boldsymbol{m}') \subset pairs(\boldsymbol{m}^\star)$ is not a qualified rival motif, more rival motifs will be required as evidence for undesignability.

Theorem 4. *If* $\exists M = \{\boldsymbol{m}_1, \boldsymbol{m}_2, .., \boldsymbol{m}_k\}$ *and* $\boldsymbol{m}^\star \notin M$*, such that* $\forall \boldsymbol{x}, \exists \boldsymbol{m} \in M, \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}) \leq \Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}^\star)$*, then* $\boldsymbol{m}$ *is undesignable.*

The right side of Fig. 4 shows an example of target motif with multiple rival motifs. Identifying a set of multiple rival motifs is more complicated than finding a single rival motif. We present a high-level Algorithm 1 to elucidate the fundamental procedures involved in identifying rival motifs, providing readers with an overview of the essential steps. In Algorithm 1 there are three parameters M, N, and K. They limit the number of differential positions, rival structures and sampled sequences, preventing the algorithm running forever. The overall complexity is $\mathcal{O}(NM + NK|\boldsymbol{m}|^3)$. We omit the intricate details for conciseness, and encourage readers to consult the literature of RIGEND [35] for a comprehensive description.

[1] The operator $\ominus$ denotes the symmetric difference (or XOR) between the two sets $A \ominus B = (A \setminus B) \cup (A \setminus B)$.



(a) Target motif (b) Single rival motif (c) Target motif (d) Rival motif 1 (e) Rival motif 2 (f) Rival motif 3 (g) Rival motif 4 (h) Rival motif 5

Fig. 4: Example of target motif and rival motif(s). The target motif 4a is from the structure in Fig. 1, the target motif 4c is from Eterna100 puzzle "`Mat - Elements & Sections`" as plotted in Table Fig. 3.

Algorithm 1 Rival Motifs Search Algorithm (high-level version)

$\mathcal{X}(\boldsymbol{m}^\star < \boldsymbol{m}') = \{\boldsymbol{x} \mid \Delta\Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}^\star, \boldsymbol{m}') < 0\}$ ▷ design space: excluding sequences impossible for successful design

function RivalMotifSearch($\boldsymbol{m}^\star, \boldsymbol{y}$) ▷ motif $\boldsymbol{m}^\star$ in a structure $\boldsymbol{y}$

  $\mathcal{M}_{\text{rival}} \leftarrow \varnothing$ ▷ define a set of rival motifs

  while $\bigcap_{\boldsymbol{m}' \in \mathcal{M}_{\text{rival}}} \mathcal{X}(\boldsymbol{m}^\star < \boldsymbol{m}') \neq \varnothing$ **do** ▷ design space is not empty

    if $|\mathcal{M}_{\text{rival}}| > N$ **then return** `unkown` ▷ early stop

    for $i = 1$ **to** K **do**

      Draw $\boldsymbol{x} \in \bigcap_{\boldsymbol{m}' \in \mathcal{M}_{\text{rival}}} \mathcal{X}(\boldsymbol{m}^\star < \boldsymbol{m}')$

      if $\boldsymbol{m}'' = \text{uMFE}(\boldsymbol{x}, \boldsymbol{y} \setminus \boldsymbol{m}^\star)$ **then return** `designable` ▷ found successful uMFE design

      $\boldsymbol{m}'' \leftarrow \text{MFE}(\boldsymbol{x})$ ▷ one of MFE structures in case of multiple

      if $4^{|\Delta(\boldsymbol{m}', \boldsymbol{m}^\star)|} < M$ **then** $\mathcal{M}_{\text{rival}} \leftarrow \mathcal{M}_{\text{rival}} \cup \{\boldsymbol{m}''\}$ ▷ limit the size of differential positions

  return `undesignable`

5 Rotational Invariance

5.1 Invariance of Motif Energy

In the nearest-neighbor model, the free energy change of a loop is independent of its absolute position or specific orientation within the structure. As a result, the free energy of motifs adheres to both *translational invariance* and *rotational invariance*, though not to *symmetrical invariance*. Fig. 5 shows two groups of equivalent motifs found in the Eterna100 puzzles, demonstrating rotational invariance. The motifs $\boldsymbol{m}_a$, $\boldsymbol{m}_b$, and $\boldsymbol{m}_c$ come from puzzles with IDs #60, #81, and #88, respectively, while $\boldsymbol{m}_d$ and $\boldsymbol{m}_e$ are from the puzzle with ID #86 (refer to Table 3 for detailed structures with highlighted motifs). Among them, $\boldsymbol{m}_a$ and $\boldsymbol{m}_b$ can be rotated into each other, as can $\boldsymbol{m}_d$ and $\boldsymbol{m}_e$. However, despite both containing three bulge loops, $\boldsymbol{m}_b$ and $\boldsymbol{m}_c$ cannot be rotated into one another.

To identify and represent such rotational equivalence, we propose a noval representation for motifs, referred to as *loop-pair graph*.

5.2 Loop-Pair Graph

Definition 8. *A* **loop-pair graph** *for a pseudoknot-free RNA secondary structure $\boldsymbol{y}$ is a weighted undirected graph $G(\boldsymbol{y}) = \langle V(\boldsymbol{y}), E(\boldsymbol{y})\rangle$ where each node $v \in V$ is either a loop in $\boldsymbol{y}$ or a pair in $\boldsymbol{y}$ or the pseudo-pair node r (representing 5′ and 3′ ends rather than a base pair), (i.e., $V(\boldsymbol{y}) = loops(\boldsymbol{y}) \cup pairs(\boldsymbol{y}) \cup \{r\}$) and each edge $e = (u, v, w) \in E(\boldsymbol{y})$ connects one loop node and one pair node, with the edge weight w denoting the number of unpaired bases of the segment of the loop between the connected pair and next pair according to the direction from 5′ to 3′. For each loop node v, there is an* **ordered list** *$N(v)$ of neighbor nodes.*

The loop-pair graphs of the structure and motifs in Fig. 3 is shown in Fig. 6c. We also show Fig. 6a and Fig. 6b for comparison. The advantages of loop-pair graphs include:

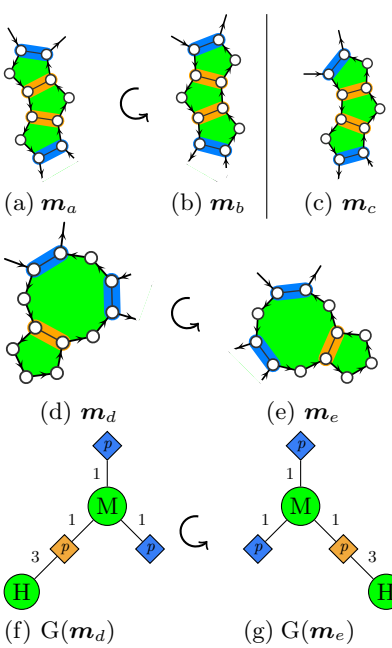


Fig. 5: Rotational invariance examples from Eterna100. $\boldsymbol{m}_a \simeq \boldsymbol{m}_b, \boldsymbol{m}_b \not\simeq \boldsymbol{m}_c, \boldsymbol{m}_d \simeq \boldsymbol{m}_e$.

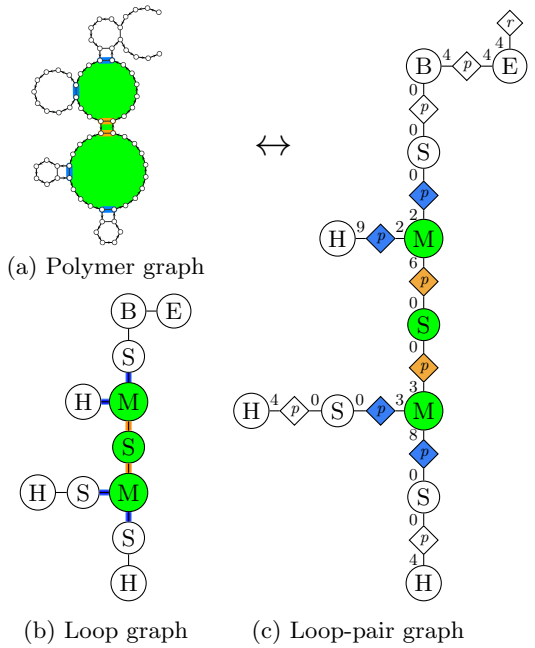


Fig. 6: Graph representations of RNA structure. A motif is a connected subgraph in a loop-pair graph (c). $\leftrightarrow$ denotes bijection.

Algorithm 2 Rotate Loop-Pair Graph via Node (see Fig. 5 for examples)

function NEWTREE(v, child_id) ▷ start recursion with a leaf node v and child_id 0
  $t \leftarrow$ TREENODE(); $p \leftarrow v$.parent ▷ create a new (sub)tree (graph) from v
  if $p \neq nil$ **then** $newchild \leftarrow$ [NEWTREE(p, v.child_id)] **else** $newchild \leftarrow []$ ▷ if v's parent exists, recursive call
  t.children $\leftarrow v$.children[child_id + 1 :] + $newchild$ + v.children[: child_id] ▷ rotate
  return t

1. **Bijective.** A loop-pair graph contains all necessary information about loops, base pairs, and unpaired bases to fully reconstruct the original structure. In contrast, *loop-graph* in Fig. 6b or RNA-as-graph representations [16,11,10,36] cannot recover the original structure due to missing information about unpaired bases or helices, respectively. ignore helix stackings and can not recover the original structure.
2. **Abstract.** Loop-pair graphs emphasize the connections and topology of loops and base pairs within motifs, while abstracting away the finer details of the backbone structure. While other representations [5,19,18] also provide abstraction of structures, they serve for different applications.
3. **Compact.** The number of unpaired bases is encoded as edge weights, which makes the representation more space-efficient than the *polymer graph* in Fig.6a.

RNA secondary structures are inherently recursive, making loop-pair graphs singly connected, essentially forming a tree. Any motif $\boldsymbol{m} \subseteq \boldsymbol{y}$ corresponds to an *induced subgraph* of $G(\boldsymbol{y})$, containing the loop nodes for each loop in $loops(\boldsymbol{m})$, the pair nodes for each pair in $pairs(\boldsymbol{m})$, and all edges connecting these nodes. To assess the *uniqueness* of motifs under rotational isomorphism, we recursively rotate the graph representation of a motif using a boundary pair node as a pivot, as described in Algorithm 2 (linear complexity).

6 Bottom-Up Scan of Motif Designability within RNA Structures

An ideal algorithm should be capable of identifying all minimal undesignable motifs within a given secondary structure. It is important to note that the sub-motifs of $\mathcal{M}$ always possess smaller cardinalities compared to $\mathcal{M}$. Therefore, a straightforward approach involves enumerating all motifs in the structure according to their cardinality and determining whether each motif is designable or undesignable, as outlined in Algorithm 3. This algorithm assumes the existence of an oracle function, DECIDE($\boldsymbol{m}$), which returns whether $\boldsymbol{m}$ is designable or undesignable. While such an ideal function DECIDE($\boldsymbol{m}$) theoretically exists (exhaustive search

Algorithm 3 Fully Bottom-Up Framework for Identifying Minimal Undesignable Motif

function BottomUpMotifDesignabilityScan($\boldsymbol{y}$) ▷ input a secondary structure $\boldsymbol{y}$
  $\mathcal{M}_{\text{miniundesignable}} \leftarrow \varnothing$ ▷ a set to store identified minimal undesignable motifs
  for all non-singleton $\boldsymbol{m} \subseteq \boldsymbol{y}$ in increasing order of $card(\boldsymbol{m})$ **do** ▷ $card(\boldsymbol{m}) = 2, 3, \ldots, |loops(\boldsymbol{y})|$
    if $\exists \boldsymbol{m}' \in \mathcal{M}_{\text{miniundesignable}}$ and $\boldsymbol{m}' \subset \boldsymbol{m}$ **then continue** ▷ undesignable but not minimal
    $designablity \leftarrow$ Decide($\boldsymbol{m}$) ▷ either `designable` or `undesignable`
    if $designablity =$ `undesignable` **then** $\mathcal{M}_{\text{miniundesignable}} \leftarrow \mathcal{M}_{\text{miniundesignable}} \cup \{\boldsymbol{m}\}$
  return $\mathcal{M}_{\text{miniundesignable}}$

Algorithm 4 FastMotif (see Fig. 7 for an example of powerset)

global: $\mathcal{M}_{\text{miniundesignable}}$, $\mathcal{M}_{\text{designable}}$ ▷ Global (minimal) undesignable and designable motifs
function FastMotif($\boldsymbol{y}$) ▷ Input is a structure
  for all loop node $u \in G(\boldsymbol{y})$ **do**
    for all non-empty $s \in 2^{N(u)}$ in increasing order of $|s|$ **do** ▷ $2^{N(u)}$: powerset of $N(u)$;$|s| = 1, 2, \ldots, |N(u)|$
      $\boldsymbol{m} \leftarrow \{u\} \cup s$ ▷ motif $\boldsymbol{m}$ has loop u and its neighbors in s, so $card(\boldsymbol{m}) \geq 2$
      if $\boldsymbol{m} \in \mathcal{M}_{\text{miniundesignable}}$ **then continue** ▷ check every rotated version of $\boldsymbol{m}$; already identified
      else
        if $\exists \boldsymbol{m}' \in \mathcal{M}_{\text{miniundesignable}}$ and $\boldsymbol{m}' \subset \boldsymbol{m}$ **then continue** ▷ undesignable but not minimal
        $designablity \leftarrow$ RivalMotifSearch($\boldsymbol{m}$) ▷ return `designable` or `undesignable` or `unkown`
        if $designablity =$ `designable` **then** $\mathcal{M}_{\text{designable}} \leftarrow \mathcal{M}_{\text{designable}} \cup \{\boldsymbol{m}\}$
        else if $designablity =$ `undesignable` **then**
          if $\forall \boldsymbol{m}_{\text{sub}} \subset \boldsymbol{m}$, $\boldsymbol{m}_{\text{sub}} \in \mathcal{M}_{\text{designable}}$ **then** $\mathcal{M}_{\text{miniundesignable}} \leftarrow \mathcal{M}_{\text{miniundesignable}} \cup \{\boldsymbol{m}\}$ ▷ minimal

could achieve this), it may not be practical due to its high complexity. Nevertheless, it serves as a conceptual foundation for developing more practical algorithms.

Theorem 5. *Given a secondary structure $\boldsymbol{y}$, Algorithm 3 outputs a set $\mathcal{M}_{miniundesignable}$ containing all and only the minimal undesignable motifs in $\boldsymbol{y}$.*

The proof is conducted by induction (detailed in Supplementary Section F.1). The total number of motifs in a structure can grow exponentially with cardinalities, making Algorithm 3 impractical for large structures. However, empirical observations suggest minimal undesignable motifs typically involve a small number of loops. To address this, we propose a variant of Algorithm 3, referred to as FastMotif (Algorithm 4), designed to identify as many minimal undesignable motifs as possible within a given structure $\boldsymbol{y}$, while maintaining computational efficiency. In particular, we limit our evaluation to motifs composed of a loop and any (non-empty) subset of its neighboring loops (Fig. 7). This approach offers several key advantages:

1. Each motif has a limited number of loops, making undesignability easier to decide.
2. For a loop v with k neighbor loops, i.e., $|N(v)| = k$, the size of $N(v)$'s power set $2^{N(v)}$, excluding $\varnothing$, is $2^k - 1$. Since most loops have 2 or 3 neighbors, the number of motifs considered remains relatively small, while still covering most of the relevant small motifs.
3. Each loop and its neighbor loops can be seen as a small subgraph on the loop-pair graph $G(\boldsymbol{y})$. Enumerating motifs in the power set would be equivalent to running Algorithm 3 on a local subgraph.

To enhance performance, we exclude motifs with more than 3 neighbor loops. The complexity of FastMotif is determined by the product of the number of motifs scanned and the complexity of Algorithm 1, making it polynominal. Additionally, FastMotif can be adapted to scan larger motifs. In Sec.8. We incorporated an undesignable *substructure* in Eterna puzzles ("Chicken feet" and "Mutated chickenfeet".) from RIGEND[35] and proved it is a minimal undesignable motif.

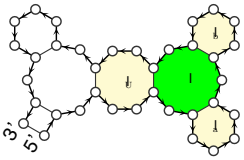


Fig. 7: Powerset in Alg. 4. Within $2^{N(l)} \setminus \varnothing$, i.e., $\{l_1\}, \{l_2\}, \{l_3\}, \{l_1, l_2\}, \{l_2, l_3\}, \{l_3, l_1\}, \{l_1, l_2, l_3\}$, only $\{l_2, l_3\}$ and $\{l_1, l_2, l_3\}$, when combined with l, are undesignable, but only $\{l, l_2, l_3\}$ is minimal undesignable (see Tab. 3 for Eterna #57).

7 Related Work

To the best of our knowledge, CountingDesign [31,30] is the existing method that has investigated undesignable motifs. However, it exhaustively enumerates and folds all RNA sequences for each motif group, identifying designable motifs and taking their complement to find undesignable ones. As a result, it is only

Table 2: Undesignable (*undes.*) structures and minimal undesignable (*m. u.*) motifs in Eterna100 puzzles & native structures from ArchiveII.

Dataset / family		*seqs.*	*structures*			*m. u. motifs*		*time per structure*
			uniq.	*avg. len.*	*undes.*	*total*	*unique*	
Eterna100 (Tab. 3)		-	100	218.3	18	36	24	59.7 s
ArchiveII	tRNA (Fig. 9)	492	175	77.1	1	1	1	0.2 s
	5S rRNA	1,147	643	118.7	23	31	17	0.5 s
	SRP	702	661	183.9	261	458	118	9.0 s
	RNaseP	429	396	332.1	99	110	60	2.1 s
	tmRNA	404	348	366.0	46	58	31	15.9 s
	Group I Intron	93	93	426.4	46	50	49	18.4 s
	telomerase	37	37	444.6	4	4	4	0.7 s
	Group II Intron	11	11	716.5	0	0	0	9.1 s
	16S rRNA	22	22	1547.9	22	79	30	502.8 s
	23S rRNA	5	5	2927.4	5	86	36	129.8 s
All		*3,342*	*2,491*	*207.6*	*525*	*913*	*370*	*8.2 s*
unique minimal undesignable motifs across all families: **355**. length: [5, 203] (avg 39.2); cardinality: [2, 5]								

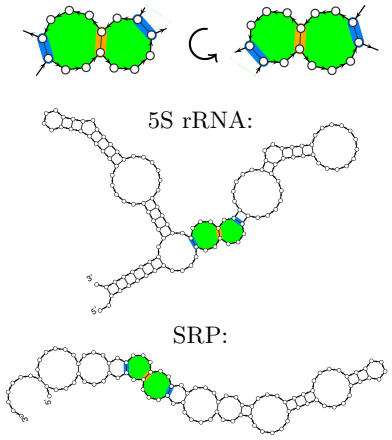


Fig. 8: A minimal undesignable motif with two rotational variants (top) found in 5S rRNA and SRP families.

applicable to short motifs, reporting motifs of up to 14 nucleotides. Another drawback is the limited interpretability of the undesignable motifs, as CountingDesign identifies them by taking the complement of designable motifs rather than directly characterizing the undesignable motifs themselves. Moreover, CountingDesign defines motifs as rooted trees starting from a base pair, making it unable to handle external loops in RNA structures. Additionally, it overlooks rotational invariance of motifs, leading to redundancy in the identified undesignable motif set. See also Sec. 8.3 and Supplementary Sec. C for efficiency comparisons.

8 Empirical Results

8.1 Settings

We applied our algorithm FastMotif to two public RNA structure benchmarks: Eterna100 [2] and ArchiveII [8]. Eterna100 consists of 100 structures[2], artificially designed by human experts, and serves as a primary benchmark for RNA design. ArchiveII, comprising native RNA structures, spans 10 families [13,25,9,7,24,38] of naturally occurring RNA and is used to evaluate RNA folding [15].[3] Following prior work in RNA design [12,22,4,34] and undesignability [35,30,31,1], the RNA folding model is ViennaRNA (v2.5.1) [20]. Our code was written in C++, utilizing a 3.40 GHz Intel Xeon E3-1231 CPU and 32 GB memory. Parallelization was achieved by OpenMP for 8 CPU cores. The parameters during rival motif search (Alg. 1) are set as $M = 10^{10}, N = 10^5, K = 100$. Our source code is available at `https://github.com/shanry/RNA-Undesign`.

8.2 Undesignable Structures and Unique Minimal Undesignable Motifs

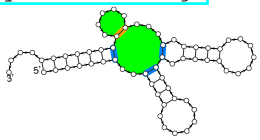


Fig. 9: Undesignable tRNA structure/motif.

Table 2 summarizes the statistics of undesignable structures and (minimal) undesignable motifs, as well as the running time, for both Eterna puzzles and native structures from ArchiveII. Among Eterna100 puzzles, we found 24 unique minimal undesignable motifs (35 occurrences), resulting in 18 undesignable puzzles (Table. 3). This result is stronger than that of RIGEND, which identified 16 undesignable puzzles along with one rival (sub-)structure for each puzzle.

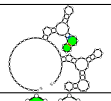
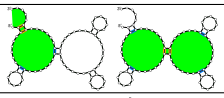
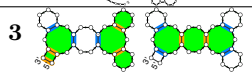
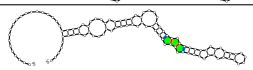
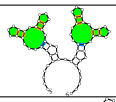
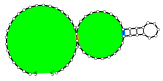
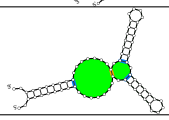
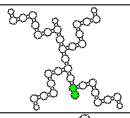
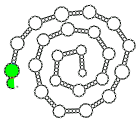
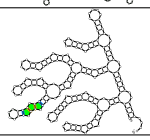
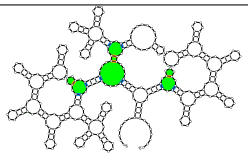
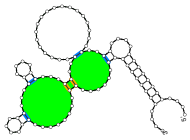
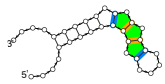
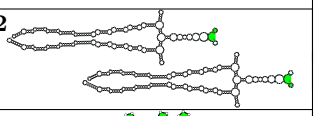
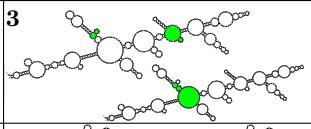
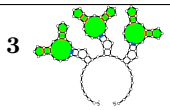
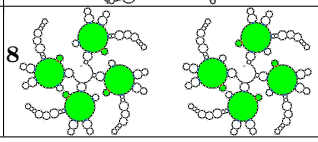
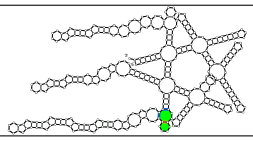
The structures in ArchiveII, on the other hand, are high-quality native structures, which intuitively should be designable. Surprisingly, there are over 1,000 occurrences of undesignable motifs from almost all ArchiveII families (except for Group II Intron). In total, we found 331 unique minimal undesignable motifs in ArchiveII, some of them shared across families, and 663 (17%) undesignable structures. For example, Fig. 8 shows a minimal undesignable motifs shared across two families and Fig. 9 shows the only undesignable tRNA structure and motif. See Supplementary Section H for more motif examples in 16S & 23S rRNA structures. We suspect the large number of undesignable structures and motifs are due to the energy model (Vienna 2) being imperfect and pseudoknots playing a role in designability which is beyond our work.

Interestingly, no motifs are shared between Eterna100 and ArchiveII, and in total we found 355 unique minimal undesignable motifs, which can be browsed on our web server `http://linearfold.org/motifs`. Users can even upload a new structure and the server will find minimal undesignable motifs in it on the fly.

[2] We used the 21 structures that do not have successfully designed sequences by unique MFE criterion

[3] Pseudoknots are removed before running our algorithm (tRNA, 5S rRNA, and SRP do not contain pseudoknots).

Table 3: Minimal Undesignable Motifs in 18 Undesignable Eterna100 Structures

ID: Puzzle	Minimal Undesignable Motifs	ID: Puzzle	Minimal Undesignable Motifs
50: 1, 2, 3 and 4 bulges[4]		52: [RNA] Repetitive Seqs. 8/10	
57: multilooping fun (see Fig. 7)		60: Mat - Elements & Sections	
61: Chicken feet		67: Simple Single Bond	
72: Loop next to a Multiloop		78: Mat - Lot 2-2 B	
80: Spiral of 5's		81: Campfire	
86: Methaqualone $C_{16}H_{14}N_2O$		87: Cat's Toy 2	
88: Zigzag Semicircle		90: Gladius	
91: Thunderbolt		92: Mutated chicken feet	
96: Cesspool		99: Shooting Star	

8.3 Efficiency

From Table 2, we can see the algorithm costs only a few seconds or minutes to scan an entire structure. Such efficiency is much superior to the previous work CountingDesign. As a comparison, identifying undesignable motifs of length 39, the average length of minimal undesignable motifs identified in our work, would take thousands of years per motif using CountingDesign (Fig. SI 2). We also apply our rival motif search algorithm (Algorithm 1) to all motifs up to length 14 to compare with CountingDesign. We successfully identified all the minimal undesignable motifs (total: 4561; unique: 1805) in 0.7 hours, whereas CountingDesign takes more than a week. See Supplementary Section C for detailed comparison with CountingDesign.

9 Conclusions and Future Work

We introduced a theoretical framework for loop-based motifs, and fast algorithms with a loop-pair graph representation to identify unique minimal undesignable motifs in RNA structures. By searching for rival motifs, the undesignability of motifs can be efficiently confirmed and explicitly explained. Future exploration could involve implementing DFS/BFS-based algorithms to search for a broader range of undesignable motifs.

The results with ArchiveII suggest the current thermodynamic parameters are deficient. We hypothesize that improvements in parameterization [37,28] could be made, particularly from the perspectives of undesignability. Future work could involve comparing the sets of minimal undesignable motifs using alternative parameter sets beyond those implemented in the ViennaRNA package, including comprehensive parameterizations that account for coaxial stacking or that are informed by experimentally known structures [29,3,23].

[4] This motif is proven undesignable if we ignore heuristic energies of special hairpins.

References

1. Aguirre-Hernández, R., Hoos, H.H., Condon, A.: Computational RNA secondary structure design: empirical complexity and improved methods. BMC bioinformatics **8**(1), 1–16 (2007)
2. Anderson-Lee, J., Fisker, E., Kosaraju, V., Wu, M., Kong, J., Lee, J., Lee, M., Zada, M., Treuille, A., Das, R.: Principles for predicting RNA secondary structure design difficulty. Journal of molecular biology **428**(5), 748–757 (2016)
3. Andronescu, M., Condon, A., Hoos, H.H., Mathews, D.H., Murphy, K.P.: Computational approaches for rna energy parameter estimation. RNA **16**(12), 2304–2318 (2010)
4. Bellaousov, S., Kayedkhordeh, M., Peterson, R.J., Mathews, D.H.: Accelerated RNA secondary structure design using preselected sequences for helices and loops. RNA **24**(11), 1555–1567 (2018)
5. Benedetti, G., Morosetti, S.: A graph-topological approach to recognition of pattern and similarity in rna secondary structures. Biophysical chemistry **59**(1-2), 179–184 (1996)
6. Bonnet, É., Rzazewski, P., Sikora, F.: Designing RNA secondary structures is hard. Journal of Computational Biology **27**(3), 302–316 (2020)
7. Brown, J.W.: The ribonuclease p database. Nucleic acids research **26**(1), 351–352 (1998)
8. Cannone, J.J., Subramanian, S., Schnare, M.N., Collett, J.R., D'Souza, L.M., Du, Y., Feng, B., Lin, N., Madabusi, L.V., Müller, K.M., Pande, N., Shang, Z., Yu, N., Gutell, R.R.: The Comparative RNA Web (CRW) Site: An Online Database of Comparative Sequence and Structure Information for Ribosomal, Intron, and Other RNAs. BioMed Central Bioinformatics **3**(2) (2002)
9. Damberger, S.H., Gutell, R.R.: A comparative database of group i intron structures. Nucleic Acids Research **22**(17), 3508–3510 (1994)
10. Gan, H.H., Fera, D., Zorn, J., Shiffeldrim, N., Tang, M., Laserson, U., Kim, N., Schlick, T.: Rag: Rna-as-graphs database—concepts, analysis, and features. Nutrition and Health **5**(1-2), 1285–1291 (1987)
11. Gan, H.H., Pasquali, S., Schlick, T.: Exploring the repertoire of rna secondary motifs using graph theory; implications for rna design. Nucleic acids research **31**(11), 2926–2943 (2003)
12. Garcia-Martin, J.A., Clote, P., Dotu, I.: RNAiFOLD: a constraint programming algorithm for RNA inverse folding and molecular design. Journal of bioinformatics and computational biology **11**(02), 1350001 (2013)
13. Gutell, R.R.: Collection of small subunit (16s-and 16s-like) ribosomal rna structures: 1994. Nucleic Acids Research **22**(17), 3502–3507 (1994)
14. Haleš, J., Maňuch, J., Ponty, Y., Stacho, L.: Combinatorial RNA design: designability and structure-approximating algorithm. In: Combinatorial Pattern Matching: 26th Annual Symposium, CPM 2015, Ischia Island, Italy, June 29–July 1, 2015, Proceedings. pp. 231–246. Springer (2015)
15. Huang, L., Zhang, H., Deng, D., Zhao, K., Liu, K., Hendrix, D.A., Mathews, D.H.: Linearfold: linear-time approximate rna folding by 5'-to-3'dynamic programming and beam search. Bioinformatics **35**(14), i295–i304 (2019)
16. Kim, N., Shiffeldrim, N., Gan, H.H., Schlick, T.: Candidates for novel rna topologies. Journal of molecular biology **341**(5), 1129–1144 (2004)
17. Koodli, R.V., Rudolfs, B., Wayment-Steele, H.K., Designers, E.S., Das, R.: Redesigning the EteRNA100 for the Vienna 2 folding engine. BioRxiv pp. 2021–08 (2021)
18. Le, S.Y., Nussinov, R., Maizel, J.V.: Tree graphs of rna secondary structures and their comparisons. Computers and Biomedical Research **22**(5), 461–473 (1989)
19. Leontis, N.B., Lescoute, A., Westhof, E.: The building blocks and motifs of rna architecture. Current opinion in structural biology **16**(3), 279–287 (2006)
20. Lorenz, R., Bernhart, S.H., Zu Siederdissen, C.H., Tafer, H., Flamm, C., Stadler, P.F., Hofacker, I.L.: ViennaRNA Package 2.0. Algorithms for Molecular Biology **6**(1), 1 (2011)
21. Mathews, D.H., Disney, M.D., Childs, J.L., Schroeder, S.J., Zuker, M., Turner, D.H.: Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proceedings of the National Academy of Sciences U.S.A. **101**(19), 7287–7292 (2004)
22. Portela, F.: An unexpectedly effective Monte Carlo technique for the RNA inverse folding problem. BioRxiv p. 345587 (2018)
23. Rivas, E., Lang, R., Eddy, S.R.: A range of complex probabilistic models for rna secondary structure prediction that includes the nearest-neighbor model and more. RNA **18**(2), 193–212 (2012)
24. Sprinzl, M., Horn, C., Brown, M., Ioudovitch, A., Steinberg, S.: Compilation of trna sequences and sequences of trna genes. Nucleic acids research **26**(1), 148–153 (1998)
25. Szymanski, M., Specht, T., Barciszewska, M.Z., Barciszewski, J., Erdmann, V.A.: 5s rrna data bank. Nucleic acids research **26**(1), 156–159 (1998)
26. Turner, D.H., Mathews, D.H.: NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. Nucleic Acids Research **38**(suppl_1), D280–D282 (2010)
27. Ward, M., Courtney, E., Rivas, E.: Fitness Functions for RNA Structure Design. bioRxiv (2022)

28. Ward, M., Sun, H., Datta, A., Wise, M., Mathews, D.H.: Determining parameters for non-linear models of multi-loop free energy change. Bioinformatics **35**(21), 4298–4306 (2019)
29. Wayment-Steele, H.K., Kladwang, W., Strom, A.I., Lee, J., Treuille, A., Becka, A., Participants, E., Das, R.: Rna secondary structure packages evaluated and improved by high-throughput experiments. Nature Methods **19**(10), 1234–1242 (2022)
30. Yao, H.T.: Local decomposition in RNA structural design. Ph.D. thesis, McGill University (Canada) (2021)
31. Yao, H.T., Chauve, C., Regnier, M., Ponty, Y.: Exponentially few RNA structures are designable. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. pp. 289–298 (2019)
32. Zadeh, J.N., Wolfe, B.R., Pierce, N.A.: Nucleic Acid Sequence Design via Efficient Ensemble Defect Optimization. Journal of Computational Chemistry **32**(3), 439–452 (2010)
33. Zhang, B., Farwell, M.: microrna: a new emerging class of players for disease diagnostics and gene therapy. Journal of cellular and molecular medicine **12**(1), 3–21 (2008)
34. Zhou, T., Dai, N., Li, S., Ward, M., Mathews, D.H., Huang, L.: RNA design via structure-aware multifrontier ensemble optimization. Bioinformatics **39**(Supplement_1), i563–i571 (2023)
35. Zhou, T., Tang, W.Y., Mathews, D.H., Huang, L.: Undesignable RNA Structure Identification via Rival Structure Generation and Structure Decomposition. To appear in Proceedings of RECOMB 2024 (2024), https://arxiv.org/pdf/2311.08339.pdf
36. Zorn, J., Gan, H.H., Shiffeldrim, N., Schlick, T.: Structural motifs in ribosomal rnas: implications for rna design and genomics. Biopolymers: Original Research on Biomolecules **73**(3), 340–347 (2004)
37. Zuber, J., Mathews, D.H.: Estimating uncertainty in predicted folding free energy changes of rna secondary structures. RNA **25**(6), 747–754 (2019)
38. Zwieb, C., Wower, J.: tmrdb (tmrna database). Nucleic acids research **28**(1), 169–170 (2000)

Supplementary Information

Tianshuo Zhou[1], Wei Yu Tang[1,6], Apoorv Malik[1], David H. Mathews[3,4,5], and Liang Huang[1,2]

[1] School of EECS and [2] Dept. of Biochemistry & Biophysics, Oregon State University, Corvallis OR 97330, USA
[3] Dept. of Biochemistry & Biophysics, [4] Center for RNA Biology, and [5] Dept. of Biostatistics & Computational Biology, University of Rochester Medical Center, Rochester, NY 14642, USA
[6] Department of Quantitative and Computational Biology, University of Southern California, CA 90089, USA

A Loops in RNA Structures

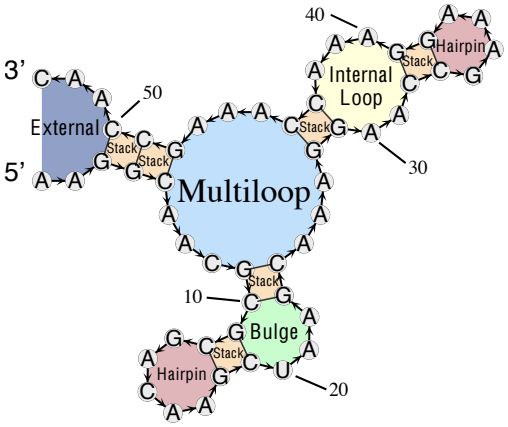


Fig. SI 1: An example of secondary structure and loops.

Table SI 1: Critical positions of loops in Fig. 2

Loop Type	Critical Positions	
	Closing Pairs	Mismatches (Unpaired)
External	(3, 50)	2, 51
Stack	(3, 50), (4, 49)	
Stack	(4, 49), (5, 48)	
Multi	(5, 48), (9, 24), (28, 44)	4, 49, 8, 25, 27, 45
Stack	(9, 24), (10, 23)	
Bulge	(10, 23), (11, 19)	
Stack	(11, 19), (12, 18)	
Hairpin	(12, 18)	13, 17
Stack	(28, 44), (29, 43)	
Internal	(29, 43), (32, 39)	30, 42, 31, 40
Stack	(32, 39), (33, 38)	
Hairpin	(33, 38)	34, 37

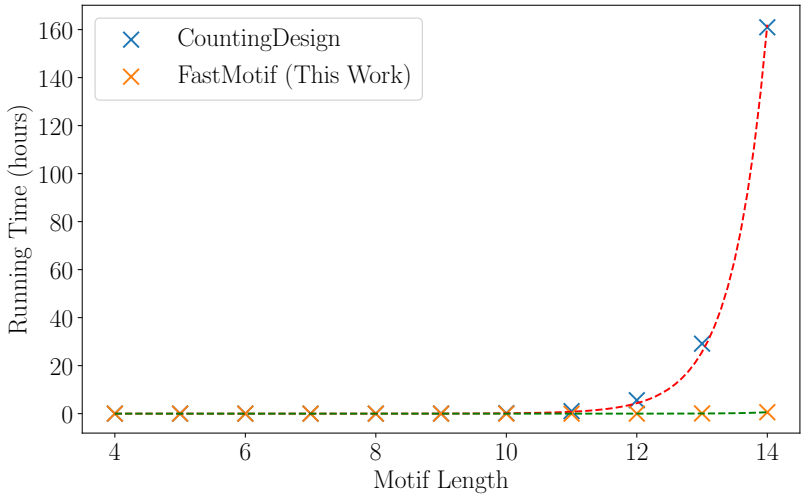
Table SI 2: Critical positions for each type of loops under Turner model implemented in ViennaRNA Special hairpins [21] of triloops, tetraloops and hexaloops are not considered here

Loop Type	Critical Positions	
	Closing Pairs	Mismatches
External	$(i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k)$	$(i_1 - 1, j_1 + 1), (i_2 - 1, j_2 + 1), \ldots, (i_k - 1, j_k + 1)$
Hairpin	(i, j)	$i + 1, j - 1$
Stack	$(i, j), (k, l)$	
Bulge	$(i, j), (k, l)$	
Internal	$(i, j), (k, l)$	$i + 1, j - 1, k - 1, l + 1$
Multi	$(i, j), (i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k)$	$i + 1, j - 1, i_1 - 1, j_1 + 1, i_2 - 1, j_2 + 1, \ldots, i_k - 1, j_k + 1$

B Projection and Intersection

Algorithm 5 Projection $\hat{\boldsymbol{x}} = \boldsymbol{x} \vdash I$

function PROJECTION($\boldsymbol{x}, I$) ▷ $I = [i_1, i_2, \ldots, i_n]$ is a list of critical positions
  $\hat{\boldsymbol{x}} \leftarrow$ map()
  for i in I **do**
    $\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{x}_i$ ▷ Project the i-th nucleotide to index i
  return $\hat{\boldsymbol{x}}$



	Cumulative Runtime
CountingDesign	1.17 weeks
FastMotif	0.7 hours

Fig. SI 2: Running time comparison between FastMotif (this work) and CountingDesign for evaluating motifs of lengths up to 14.

Algorithm 6 Contraint Intersection $C' = \text{Intersection}(C_1, C_2)$

function INTERSECTION(C_1, C_2) ▷ C_1, C_2 are sets of constraints
  $(I_1, \hat{X}_1) \leftarrow C_1$ ▷ I contains critical positions and $\hat{X}$ is a set of nucleotides compositions
  $(I_2, \hat{X}_2) \leftarrow C_2$
  $I' \leftarrow I_1 \cap I_2$
  if $I' = \varnothing$ **then** ▷ No overlapping positions; return original constraints
    return C_1, C_2
  $\hat{X}_1' \leftarrow \{\hat{\boldsymbol{x}} \vdash I' \mid \hat{\boldsymbol{x}} \in \hat{X}_1\}$
  $\hat{X}_2' \leftarrow \{\hat{\boldsymbol{x}} \vdash I' \mid \hat{\boldsymbol{x}} \in \hat{X}_2\}$
  for $\hat{\boldsymbol{x}} \in \hat{X}_1$ **do** ▷ Remove nucleotides compositions from $\hat{X}_1$ that is not in $\hat{X}_2$
    if $\hat{\boldsymbol{x}} \vdash I' \notin \hat{X}_2'$ **then** $\hat{X}_1 \leftarrow \hat{X}_1 \setminus \{\hat{\boldsymbol{x}}\}$
  for $\hat{\boldsymbol{x}} \in \hat{X}_2$ **do** ▷ Remove nucleotides compositions from $\hat{X}_2$ that is not in $\hat{X}_1$
    if $\hat{\boldsymbol{x}} \vdash I' \notin \hat{X}_1'$ **then** $\hat{X}_2 \leftarrow \hat{X}_2 \setminus \{\hat{\boldsymbol{x}}\}$
  $C_1' \leftarrow (I_1,\ \hat{X}_1)$
  $C_2' \leftarrow (I_2,\ \hat{X}_2)$
  return $C_1' \cup C_2'$ ▷ Return updated constraints

C Comparison with CountingDesign

We compared FastMotif with the original CountingDesign [31] program[1] to identify all undesignable motifs up to 14 nucleotides in length under the same setting (3.40 GHz Intel Xeon E3-1231 CPU and 32 GB of memory, with parallelization of 8 CPU cores). Fig. SI 2 shows the total running times of the two methods for identifying undesignable motifs of different lengths. Both methods found all all the undesignable motifs (total: 4561; unique: 1805) and the designable motifs up to length of 14. However, the time cost of CountingDesign increases exponentially with motif lengths, highlighting its impracticality for longer motifs. In contrast, FastMotif only need 0.7 hours. More importantly, FastMotif identified a set of rival motifs for each undesignable motif, which is explainable and helpful for further understanding RNA folding. Therefore, FastMotif demonstrates significant advantages in terms of not only scalability but also interpretability.

[1] https://gitlab.com/htyao/countingdesign

D Brute-Force Enumeration and Folding

Given a target motif $\boldsymbol{m}^\star \subseteq \boldsymbol{y}$, the most straightforward method is to enumerate all possible nucleotides compositions , and check whether there exists at least one composition that can fold into $\boldsymbol{m}^\star$ under the constraint $\boldsymbol{y} \setminus \boldsymbol{m}^\star$. However, this is impractical in reality because of high time cost. The design for $\boldsymbol{m}^\star$ should at least satisfy that nucleotides at the paired position should be matchable, the number of brute-force enumeration is $6^{|pairs(\boldsymbol{m}^\star)|} \times 4^{|unpaired(\boldsymbol{m}^\star)|}$, as there are 6 choices for a pair and 4 types of nucleotides. Constrained folding algorithms typically have a cubic time complexity with respect to length, the overall complexity

$$\mathcal{O}(6^{|pairs(\boldsymbol{m}^\star)|} \times 4^{|unpaired(\boldsymbol{m}^\star)|} \cdot (2|pairs(\boldsymbol{m}^\star)| + |unpaired(\boldsymbol{m}^\star)|)^3)$$

makes exhaustive search impractical even for small structures.

E Single Rival Motif Property

Suppose there exists a pair (i, j) such that $(i, j) \in pairs(\boldsymbol{m}')$ but $(i, j) \notin pairs(\boldsymbol{m}^\star)$. For any sequence $\boldsymbol{x}$ where $\boldsymbol{x}_i\boldsymbol{x}_j$ is not among allowed to pair, i.e. $\boldsymbol{x}_i\boldsymbol{x}_j \notin \{\mathrm{CG}, \mathrm{GC}, \mathrm{AU}, \mathrm{UA}, \mathrm{GU}, \mathrm{UG}\}$, $\boldsymbol{x}$ cannot fold into $\boldsymbol{m}'$ because $\Delta G^\circ(\boldsymbol{x}, \boldsymbol{m}') = \infty$. Therefore, if $\boldsymbol{x}$ prefers $\boldsymbol{m}'$ to $\boldsymbol{m}^\star$, then $\boldsymbol{m}'$ cannot have any pair (i, j) not in $pairs(\boldsymbol{m}^\star)$. Since $\boldsymbol{m}' \neq \boldsymbol{m}^\star$, it follows that $pairs(\boldsymbol{m}') \subset pairs(\boldsymbol{m}^\star)$.

F Proof

F.1 Proof of Theorem 5

Proof. The proof can be conducted by induction.

1. Base case. At the iteration of $i = 2$, the found undesignable motifs at line 6 are minimal because of Theorem 2. At the end of the iteration, $\mathcal{M}$ consists of all minimal undesignable motifs of 2 loops.
2. Induction hypothesis. Suppose when $i = k \geq 2$, the found undesignable motifs at line 6 are minimal undesignable motifs, and all minimal undesignable motifs of cardinality equal or less than i are included in $\mathcal{M}_{miniundesignable}$ at the end of the iteration.
3. Induction step. When $i = k + 1$, each $\boldsymbol{m}$ satisfying $|loops(\boldsymbol{m})| = k + 1$ will be checked. If $\boldsymbol{m}$ contains other undesignable motifs, line 4 will stop it from being further considered. If $\boldsymbol{m}$ is designable, line 6 will prevent it from being added to $\mathcal{M}_{miniundesignable}$. As a result, $\boldsymbol{m}$ will be added to $\mathcal{M}_{miniundesignable}$ if and only if $\boldsymbol{m}$ is minimal undesignable.

□

F.2 Proof of Theorem 2

Proof. If $|loops(\boldsymbol{m}^\star)| = 1$, then there is no internal pairs, $ipairs(\boldsymbol{m}^\star) = \varnothing$. Let each paired position in $\boldsymbol{m}^\star$ have a base pair (C, G) or (G, C) and unpaired position in $\boldsymbol{m}^\star$ have a nucleotide A, then no internal pair can be formed. $\boldsymbol{m}^\star$ is the only possible motif in the motif ensemble of constrained folding, i.e., $\mathcal{M}(\boldsymbol{x}, \boldsymbol{y} \backslash \boldsymbol{m}^\star) = \{\boldsymbol{m}^\star\}$ □

G Energy Scoring Functions

G.1 Hairpin

For the list of special hairpins, see Table SI 3. Set length $= j - i - 1$.

$$\Delta G^\circ(\boldsymbol{x}, H\langle(i,j)\rangle) = \begin{cases} \Delta G^\circ_{\text{special\_hairpin}}(\boldsymbol{x}) & \text{if } (\boldsymbol{x}, H\langle(i,j)\rangle) \text{ is special hairpin;} \\ \left(\begin{array}{l} \Delta G^\circ_{\text{hairpin\_mismatch}}((\boldsymbol{x}_i, \boldsymbol{x}_j), \boldsymbol{x}_{i+1}, \boldsymbol{x}_{j-1}) \\ + \begin{cases} \Delta G^\circ_{\text{hairpin}}(\text{length}) & \text{if } 0 \leq \text{length} \leq 30 \\ \Delta G^\circ_{\text{hairpin}}(30) + \Delta G^\circ_{\text{extend}} \cdot \ln((\text{length})/30) & \text{if length} > 30 \end{cases} \end{array}\right) & \text{otherwise.} \end{cases}$$

Table SI 3: List of special hairpins in the Turner energy model.

$\boldsymbol{x}$	$\Delta G^\circ_{\text{special\_hairpin}}(\boldsymbol{x}, (\ldots))$
CAACG	6.8
GUUAC	6.9

$\boldsymbol{x}$	$\Delta G^\circ_{\text{special\_hairpin}}(\boldsymbol{x}, (\ldots.))$
CAACGG	5.5
CCAAGG	3.3
CCACGG	3.7
CCCAGG	3.4
CCGAGG	3.5
CCGCGG	3.6
CCUAGG	3.7
CCUCGG	2.5
CUAAGG	3.6
CUACGG	2.8
CUCAGG	3.7
CUCCGG	2.7
CUGCGG	2.8
CUUAGG	3.5
CUUCGG	3.7

$\boldsymbol{x}$	$\Delta G^\circ_{\text{special\_hairpin}}(\boldsymbol{x}, (\ldots\ldots))$
ACAGUACU	2.8
ACAGUGCU	2.9
ACAGUGAU	3.6
ACAGUUCU	1.8

G.2 Special Hairpin

See Table SI 3.

G.3 Stack

$$\Delta G^\circ(\boldsymbol{x}, S\langle(i,j),(k,l)\rangle) = \Delta G^\circ_{\text{stack}}((\boldsymbol{x}_i, \boldsymbol{x}_j), (\boldsymbol{x}_k, \boldsymbol{x}_l))$$

G.4 Bulge

Set length $= k - i + j - l - 2$.

$$\Delta G^\circ(\boldsymbol{x}, B\langle(i,j),(k,l)\rangle) = \begin{cases} \Delta G^\circ_{\text{bulge}}(\text{length}) & \text{if length} \le 30 \\ \Delta G^\circ_{\text{bulge}}(30) + \Delta G^\circ_{\text{extend}} \cdot \ln(\text{length}/30) & \text{if length} > 30 \end{cases}$$
$$+ \begin{cases} \Delta G^\circ_{\text{stack}}((\boldsymbol{x}_i, \boldsymbol{x}_j), (\boldsymbol{x}_k, \boldsymbol{x}_l)) & \text{if length} = 1 \\ \left(\begin{cases} \Delta G^\circ_{\text{AU/GU}} & \text{if } \boldsymbol{x}_i = \text{U or } \boldsymbol{x}_j = \text{U} \\ 0 & \text{otherwise} \end{cases} + \begin{cases} \Delta G^\circ_{\text{AU/GU}} & \text{if } \boldsymbol{x}_k = \text{U or } \boldsymbol{x}_l = \text{U} \\ 0 & \text{otherwise} \end{cases} \right) & \text{otherwise} \end{cases}$$

G.5 Internal Loop

For 1×1, 1×2, 2×1, 2×2, 2×3, and $1 \times n$ special internal loops, see Supplementary Section G.6. Set length $= (k - i) + (j - l) - 2$.

$$\begin{aligned} \Delta G^\circ(\boldsymbol{x}, I\langle(i,j),(k,l)\rangle) = {} & \Delta G^\circ_{\text{internal\_mismatch}}((\boldsymbol{x}_i, \boldsymbol{x}_j), \boldsymbol{x}_{i+1}, \boldsymbol{x}_{j-1}) \\ & + \Delta G^\circ_{\text{internal\_mismatch}}((\boldsymbol{x}_k, \boldsymbol{x}_l), \boldsymbol{x}_{k-1}, \boldsymbol{x}_{l+1}) \\ & + \min(n_{37}^{\max}, n_{37} \cdot |(k-i) - (j-l)|) \\ & + \begin{cases} \Delta G^\circ_{\text{internal}}(\text{length}) & \text{if length} \le 30 \\ \Delta G^\circ_{\text{internal}}(30) + \Delta G^\circ_{\text{extend}} \cdot \ln([\text{length}]/30) & \text{if length} > 30 \end{cases} \end{aligned}$$

G.6 Special Internal Loop

Set length $= k - i + j - l - 1$,

$$\Delta G^\circ_{\text{special\_internal}}(\boldsymbol{x}, I\langle(i,j),(k,l)\rangle) =$$
$$\begin{cases}
\Delta G^\circ_{\text{internal_11}}((\boldsymbol{x}_i,\boldsymbol{x}_j),(\boldsymbol{x}_k,\boldsymbol{x}_l),\boldsymbol{x}_{i+1},\boldsymbol{x}_{j-1}) & \text{if } 1\times 1\\
\Delta G^\circ_{\text{internal_12}}((\boldsymbol{x}_i,\boldsymbol{x}_j),(\boldsymbol{x}_k,\boldsymbol{x}_l),\boldsymbol{x}_{i+1},\boldsymbol{x}_{k-1},\boldsymbol{x}_{j-1}) & \text{if } 1\times 2\\
\Delta G^\circ_{\text{internal_21}}((\boldsymbol{x}_i,\boldsymbol{x}_j),(\boldsymbol{x}_k,\boldsymbol{x}_l),\boldsymbol{x}_{i+1},\boldsymbol{x}_{l+1},\boldsymbol{x}_{j-1}) & \text{if } 2\times 1\\
\Delta G^\circ_{\text{internal_22}}((\boldsymbol{x}_i,\boldsymbol{x}_j),(\boldsymbol{x}_k,\boldsymbol{x}_l),\boldsymbol{x}_{i+1},\boldsymbol{x}_{k-1}\boldsymbol{x}_{l+1},\boldsymbol{x}_{j-1}) & \text{if } 2\times 2\\
\left(\begin{array}{l}\Delta G^\circ_{\text{internal}}(5) + n_{37}\\ + \Delta G^\circ_{\text{internal_23_mismatch}}((\boldsymbol{x}_i,\boldsymbol{x}_j),\boldsymbol{x}_{i+1},\boldsymbol{x}_{j-1})\\ + \Delta G^\circ_{\text{internal_23_mismatch}}((\boldsymbol{x}_k,\boldsymbol{x}_l),\boldsymbol{x}_{k-1},\boldsymbol{x}_{l+1})\end{array}\right) & \text{if } 2\times 3\\
\left(\begin{array}{l}\Delta G^\circ_{\text{internal_1n_mismatch}}((\boldsymbol{x}_i,\boldsymbol{x}_j),\boldsymbol{x}_{i+1},\boldsymbol{x}_{j-1})\\ + \Delta G^\circ_{\text{internal_1n_mismatch}}((\boldsymbol{x}_k,\boldsymbol{x}_l),\boldsymbol{x}_{k-1},\boldsymbol{x}_{l+1})\\ + \min(n_{37}^{\max}, n_{37}\cdot|(k-i)-(j-l)|)\\ + \begin{cases}\Delta G^\circ_{\text{internal}}(\text{length}+1) & \text{if length}+1\le 30\\ \Delta G^\circ_{\text{internal}}(30) + \Delta G^\circ_{\text{extend}}\cdot\ln([\text{length}+1]/30) & \text{if } (k-i)+(j-l)-2>30\end{cases}\end{array}\right) & \text{if } 1\times n
\end{cases}$$

G.7 Multiloop

Let $m = (i,j),(i_1,j_1),(i_2,j_2),\ldots,(i_k,j_k)$

$$\Delta G^\circ(\boldsymbol{x}, M\langle m\rangle) = \Delta G^\circ_{\text{closing\_pair}}(i,j) + \sum_{l=1}^{k} \Delta G^\circ_{\text{enclosed\_pair}}(\boldsymbol{x}, i_l, j_l)$$

$$\begin{aligned}\Delta G^\circ_{\text{closing\_pair}}(i,j) = {}& \Delta G^\circ_{\text{closing}}\\ &+ \Delta G^\circ_{\text{multi\_mismatch}}((\boldsymbol{x}_i,\boldsymbol{x}_j),\boldsymbol{x}_{i+1},\boldsymbol{x}_{j-1})\\ &+ \Delta G^\circ_{\text{multi\_internal}}\\ &+ \begin{cases}\Delta G^\circ_{\text{AU/GU}} & \text{if } \boldsymbol{x}_i = \text{U or } \boldsymbol{x}_j = \text{U}\\ 0 & \text{otherwise}\end{cases}\end{aligned}$$

$$\begin{aligned}\Delta G^\circ_{\text{enclosed\_pair}}(i,j) = {}& \Delta G^\circ_{\text{multi\_mismatch}}((\boldsymbol{x}_i,\boldsymbol{x}_j),\boldsymbol{x}_{i-1},\boldsymbol{x}_{j+1})\\ &+ \Delta G^\circ_{\text{multi\_internal}}\\ &+ \begin{cases}\Delta G^\circ_{\text{AU/GU}} & \text{if } \boldsymbol{x}_i = \text{U or } \boldsymbol{x}_j = \text{U}\\ 0 & \text{otherwise}\end{cases}\end{aligned}$$

Fig. SI 3: Side by side alignment of equations.

G.8 External Loop

$$\begin{aligned}\Delta G^\circ(\boldsymbol{x}, E\langle(5',3'),(i,j)\rangle) = {}& \Delta G^\circ_{\text{external\_mismatch}}((\boldsymbol{x}_i,\boldsymbol{x}_j),\boldsymbol{x}_{i-1},\boldsymbol{x}_{j+1})\\ &+ \Delta G^\circ_{\text{dangle\_5'}}((\boldsymbol{x}_i,\boldsymbol{x}_j),\boldsymbol{x}_{i-1}) + \Delta G^\circ_{\text{dangle\_3'}}((\boldsymbol{x}_i,\boldsymbol{x}_j),\boldsymbol{x}_{j+1})\\ &+ \begin{cases}\Delta G^\circ_{\text{AU/GU}} & \text{if } \boldsymbol{x}_i = \text{U or } \boldsymbol{x}_j = \text{U}\\ 0 & \text{otherwise}\end{cases}\end{aligned}$$

H Undesignable Motifs in 5S/16S/23S rRNA

H.1 5S rRNA Motifs and Structures

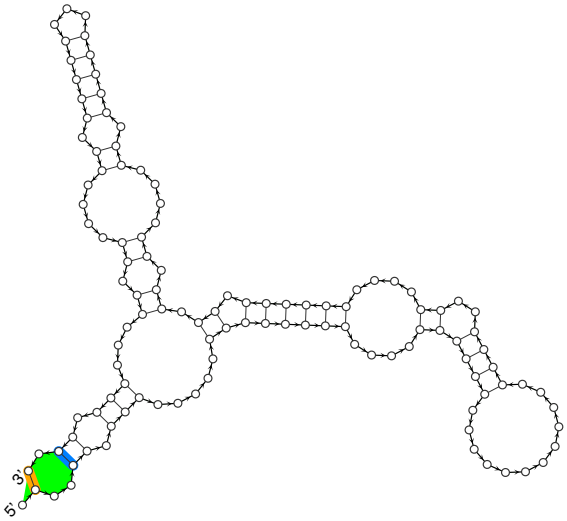


Fig. SI 4: Azospirillum-lipoferum-1, 5S rRNA

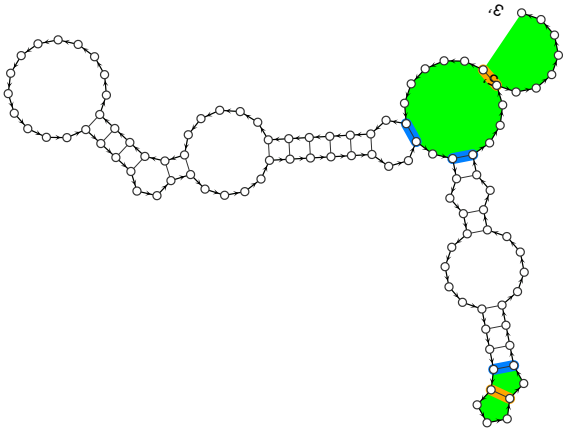


Fig. SI 5: Bacteroides-fragilis-1 & Bacteroides-fragilis-3 & Bacteroides-fragilis-4 & Bacteroides-fragilis-5, 5S rRNA

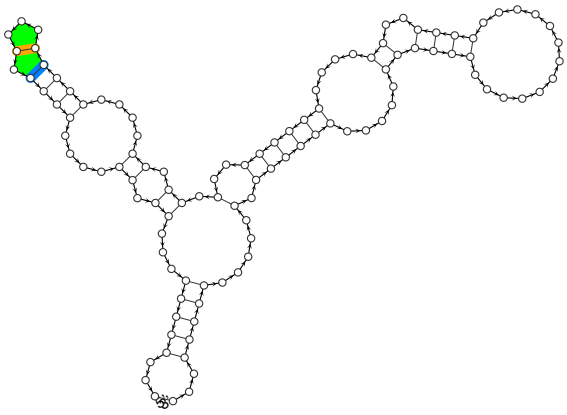


Fig. SI 6: Bacteroides-fragilis-2, 5S rRNA

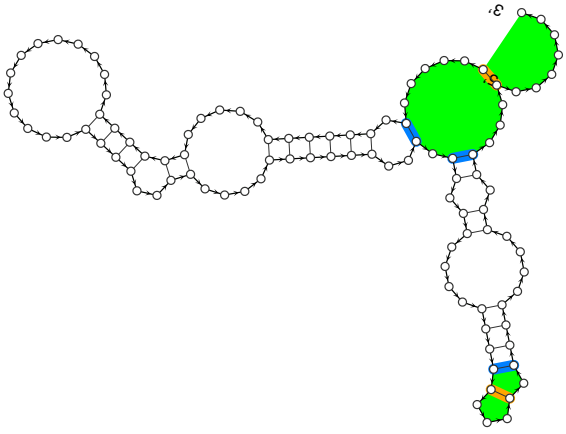


Fig. SI 7: Bacteroides-fragilis-1 & Bacteroides-fragilis-3 & Bacteroides-fragilis-4 & Bacteroides-fragilis-5, 5S rRNA

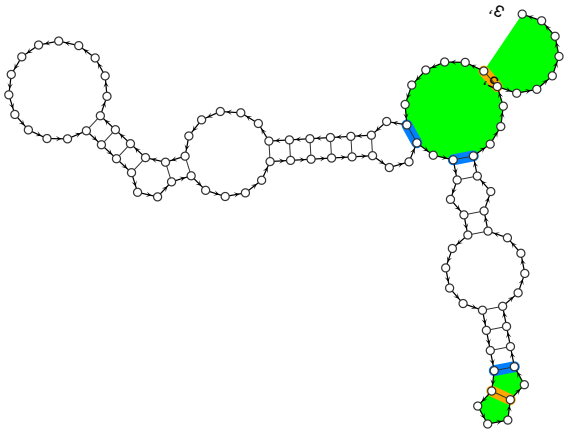


Fig. SI 8: Bacteroides-fragilis-1 & Bacteroides-fragilis-3 & Bacteroides-fragilis-4 & Bacteroides-fragilis-5, 5S rRNA

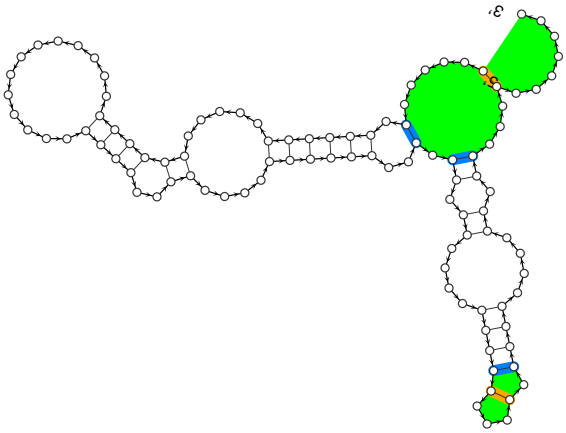


Fig. SI 9: Bacteroides-fragilis-1 & Bacteroides-fragilis-3 & Bacteroides-fragilis-4 & Bacteroides-fragilis-5, 5S rRNA

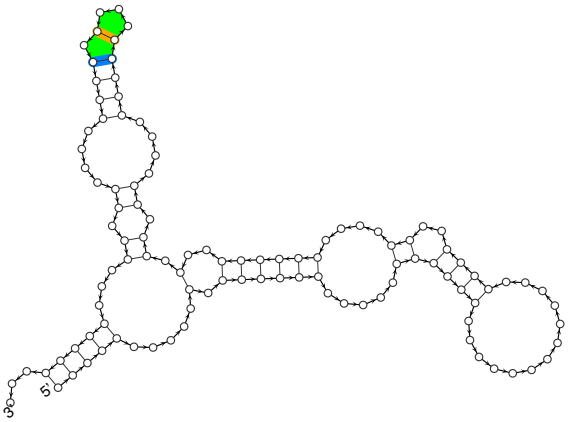


Fig. SI 10: Bacteroides-fragilis-6, 5S rRNA

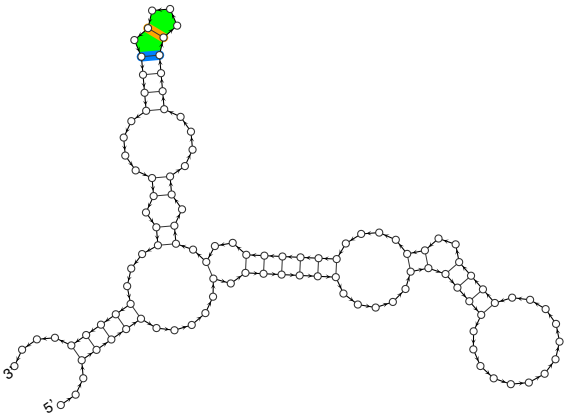


Fig. SI 11: Bacteroides-thetaiotaomicron-1, 5S rRNA

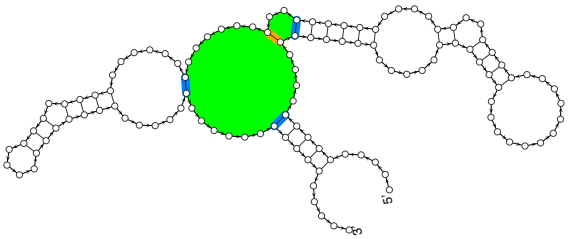


Fig. SI 12: Chlorella-pyrenoidosa-1, 5S rRNA

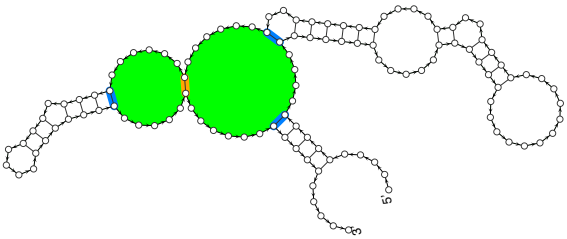


Fig. SI 13: Chlorella-pyrenoidosa-1, 5S rRNA

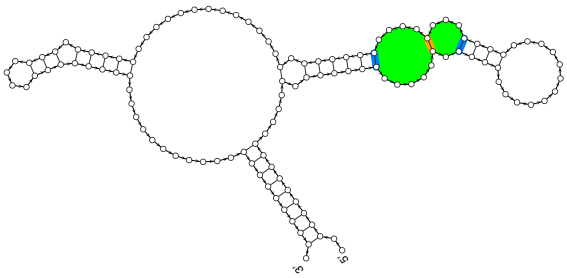


Fig. SI 14: Cryptomonas-paramecium-1, 5S rRNA

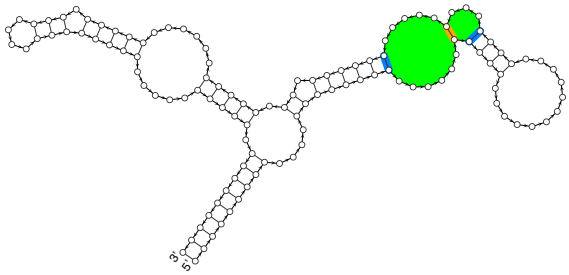


Fig. SI 15: Emericella-nidulans-4, 5S rRNA

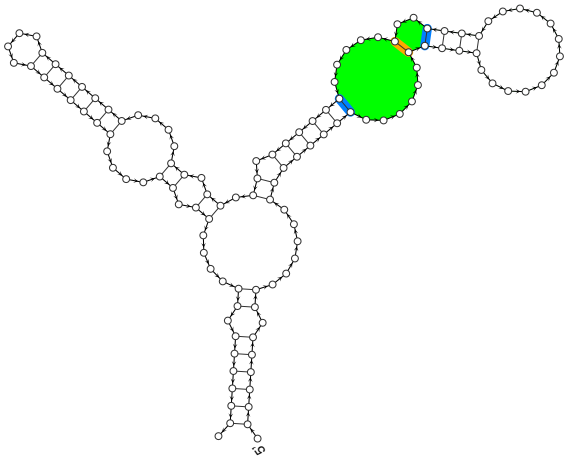


Fig. SI 16: Erythromicrobium-ezovicum-1, 5S rRNA

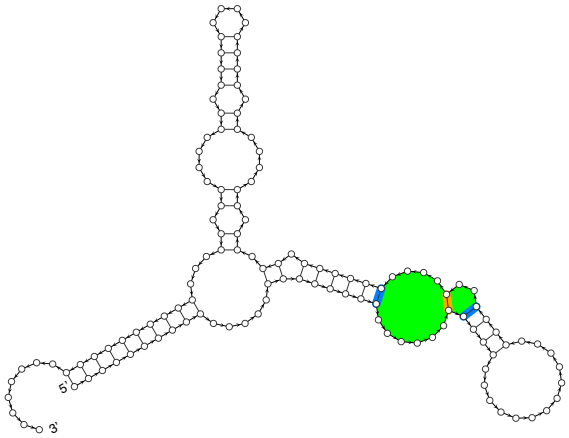


Fig. SI 17: Agrobacterium-tumefaciens-8, 5S rRNA

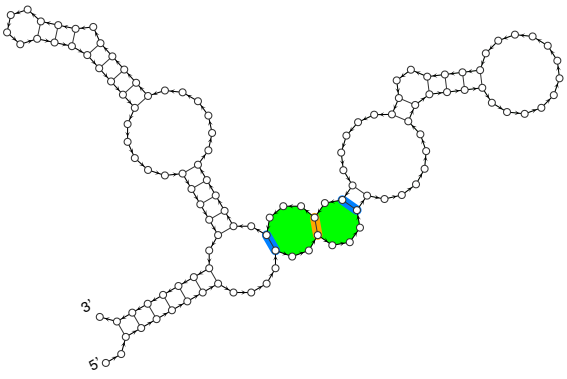


Fig. SI 18: Filobasidiella-depauperata-1, 5S rRNA

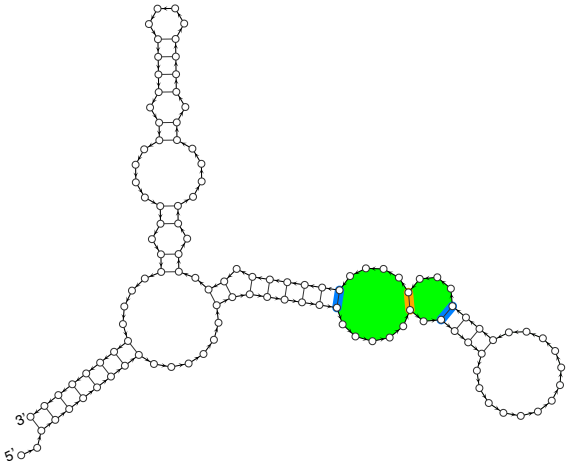


Fig. SI 19: Hyphomicrobium-vulgare-1, 5S rRNA

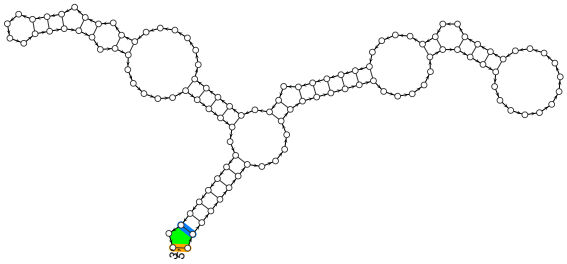


Fig. SI 20: Meloidogyne-chitwoodi-1, 5S rRNA

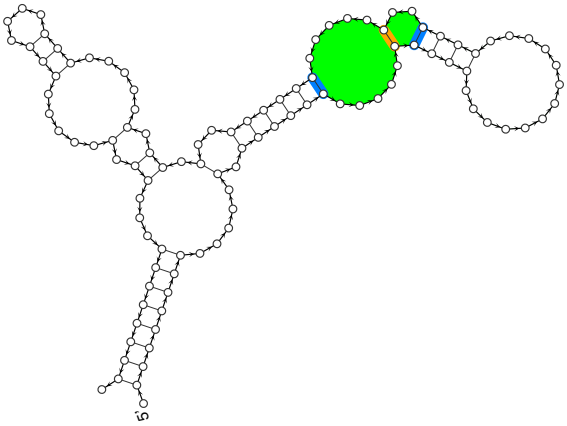


Fig. SI 21: Mycoplasma-flocculare-1, 5S rRNA

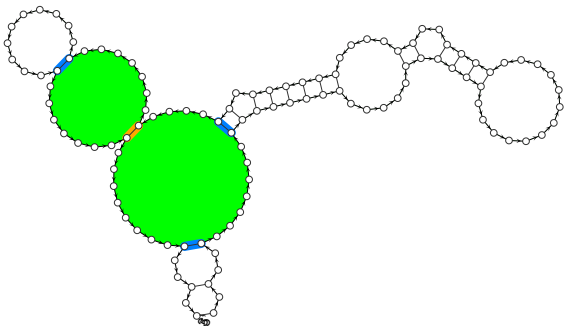


Fig. SI 22: Mycoplasma-gallisepticum-2, 5S rRNA

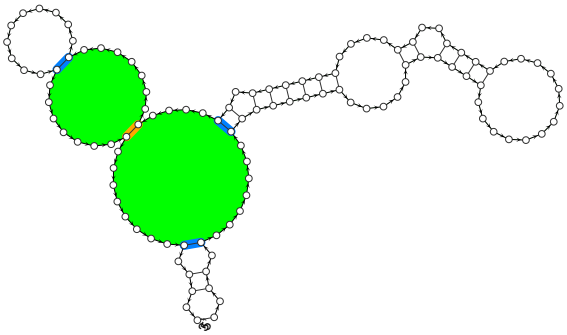


Fig. SI 23: Mycoplasma-gallisepticum-3, 5S rRNA

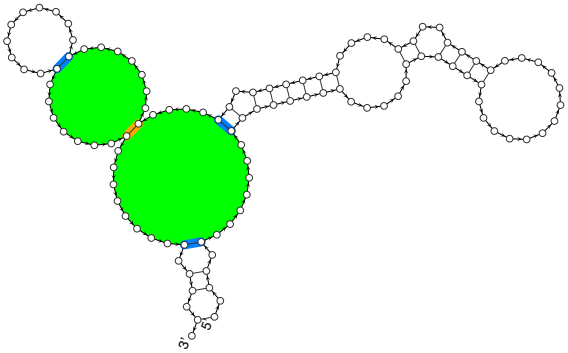


Fig. SI 24: Mycoplasma-gallisepticum-4, 5S rRNA

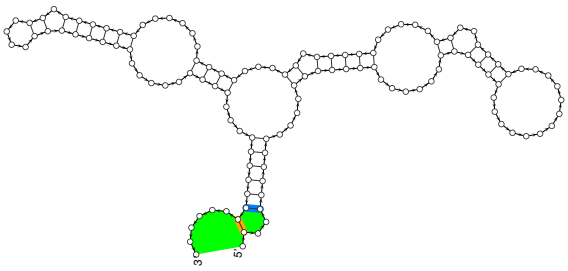


Fig. SI 25: Natrinema-sp.-1, 5S rRNA

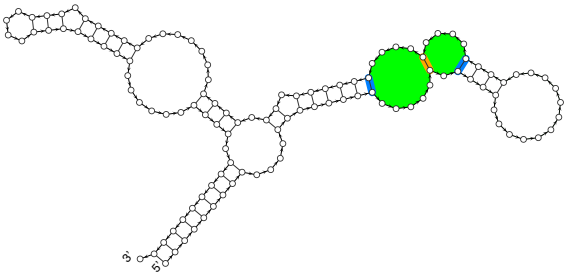


Fig. SI 26: Neurospora-crassa-11, 5S rRNA

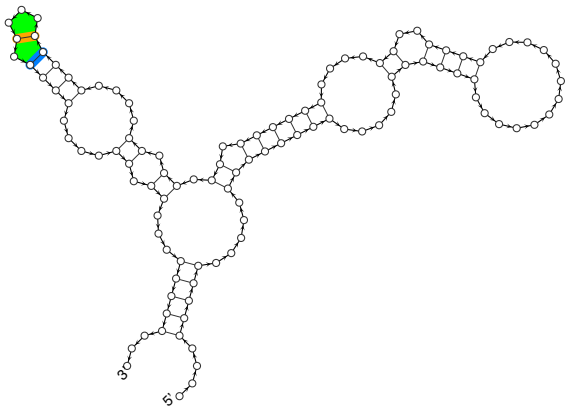


Fig. SI 27: Porphyromonas-gingivalis-1, 5S rRNA

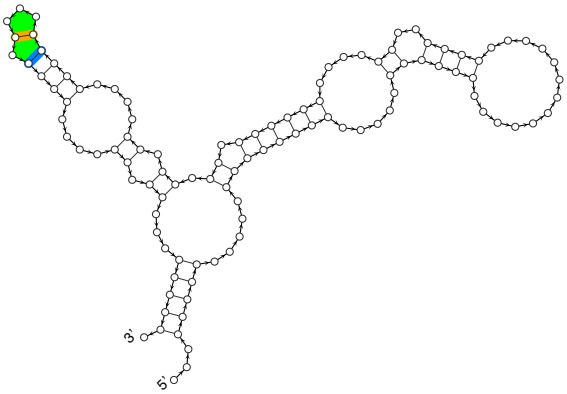


Fig. SI 28: Porphyromonas-gingivalis-2, 5S rRNA

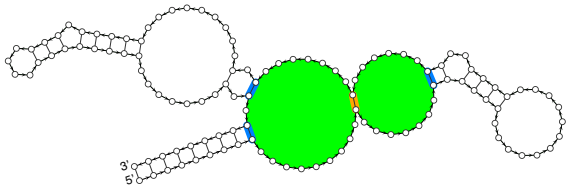


Fig. SI 29: Pyrenophora-graminea-1, 5S rRNA

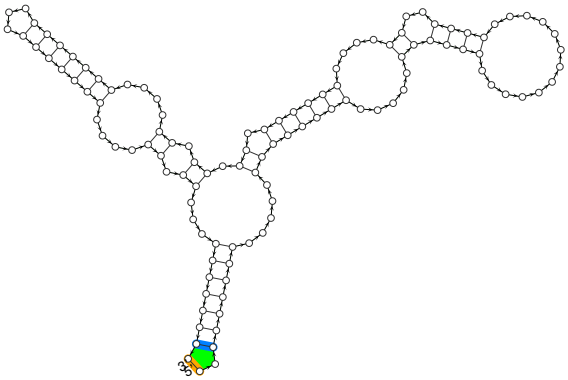


Fig. SI 30: Rubrivivax-gelatinosus-1, 5S rRNA

H.2 16S rRNA Motifs and Structures

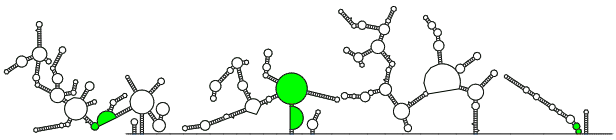


Fig. SI 31: A.fulgidus, 16S rRNA

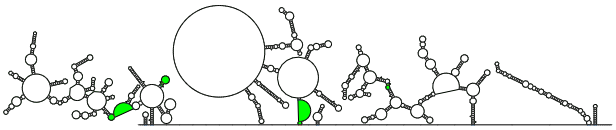


Fig. SI 32: F.ananassa, 16S rRNA

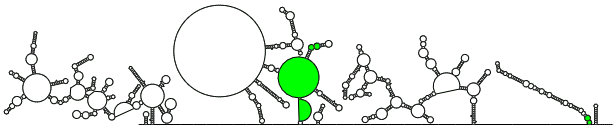


Fig. SI 33: F.ananassa, 16S rRNA

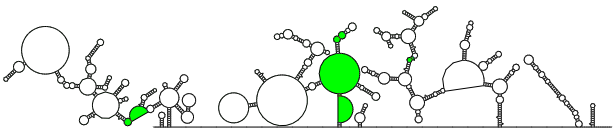


Fig. SI 34: G.intestinalis, 16S rRNA

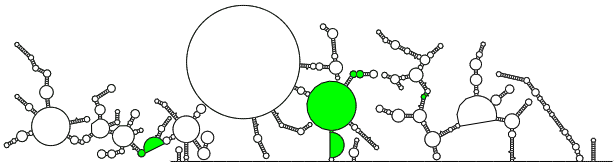


Fig. SI 35: H.sapiens, 16S rRNA

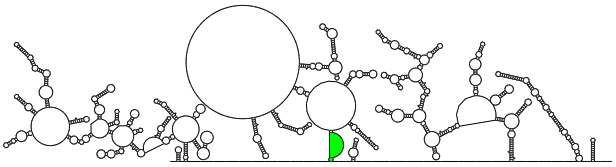


Fig. SI 36: H.sapiens, 16S rRNA

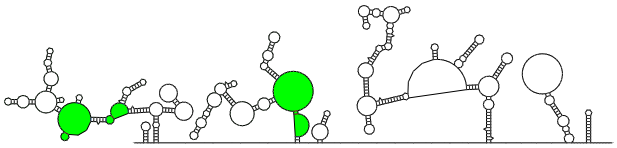


Fig. SI 37: H.sapiens.mito, 16S rRNA

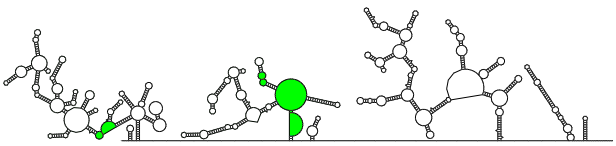


Fig. SI 38: H.volcanii, 16S rRNA

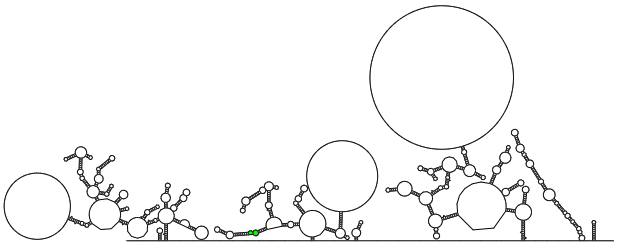


Fig. SI 39: M.polymorpha, 16S rRNA

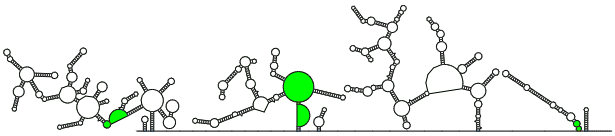


Fig. SI 40: P.occultum, 16S rRNA

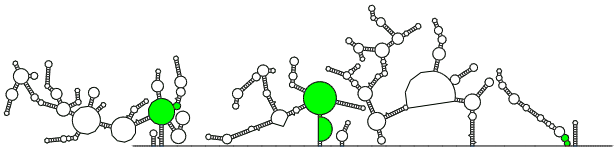


Fig. SI 41: P.staleyi, 16S rRNA

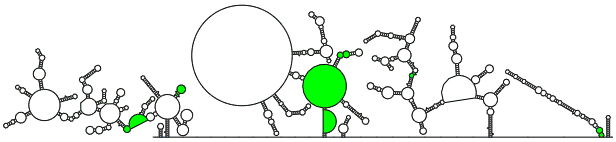


Fig. SI 42: S.cerevisiae, 16S rRNA

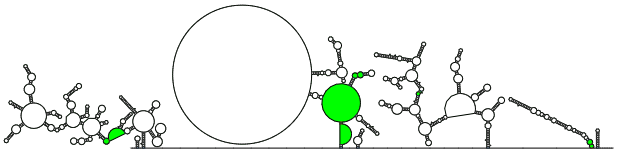


Fig. SI 43: T.gondii, 16S rRNA

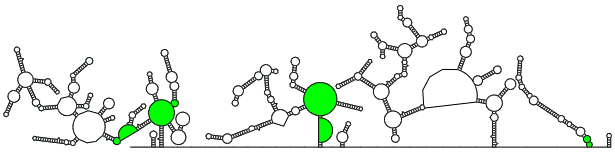


Fig. SI 44: A.pyrophilus, 16S rRNA

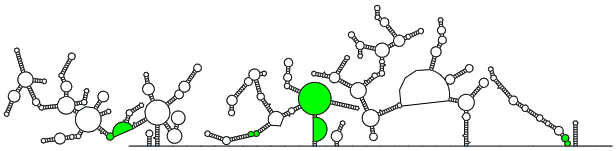


Fig. SI 45: T.maritima, 16S rRNA

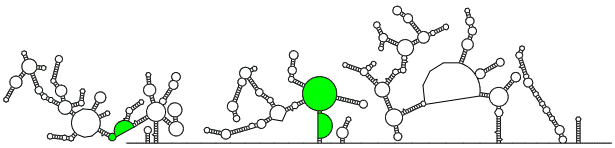


Fig. SI 46: Z.mays, 16S rRNA

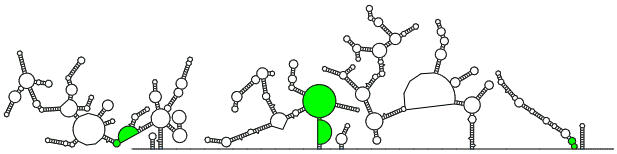


Fig. SI 47: B.burgdorferi, 16S rRNA

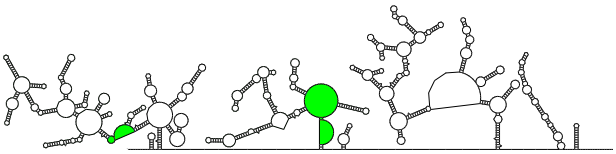


Fig. SI 48: B.subtilis, 16S rRNA

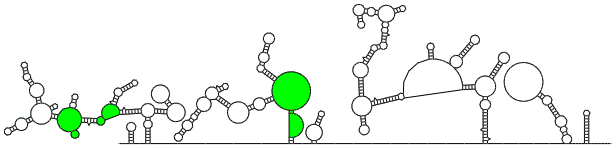


Fig. SI 49: C.elegans, 16S rRNA

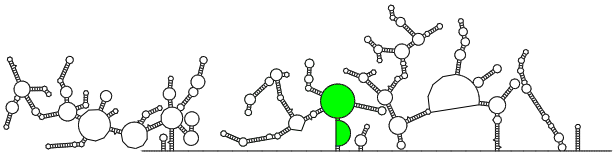


Fig. SI 50: C.psittaci, 16S rRNA

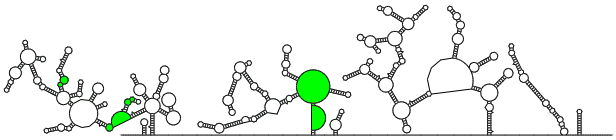


Fig. SI 51: C.reinhardtii.chloro, 16S rRNA

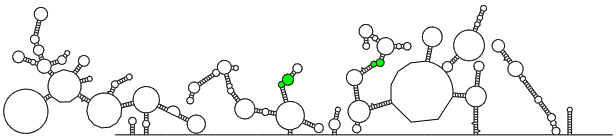


Fig. SI 52: C.reinhardtii.mito, 16S rRNA

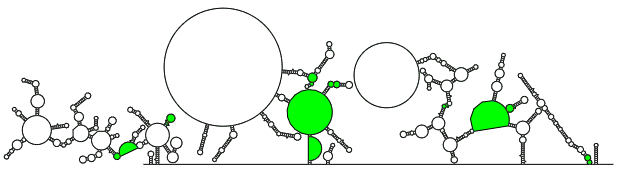


Fig. SI 53: D.melanogaster, 16S rRNA

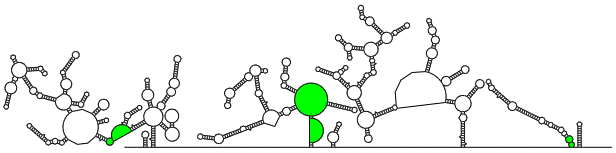


Fig. SI 54: E.coli, 16S rRNA

H.3 23S rRNA Motifs and Structures

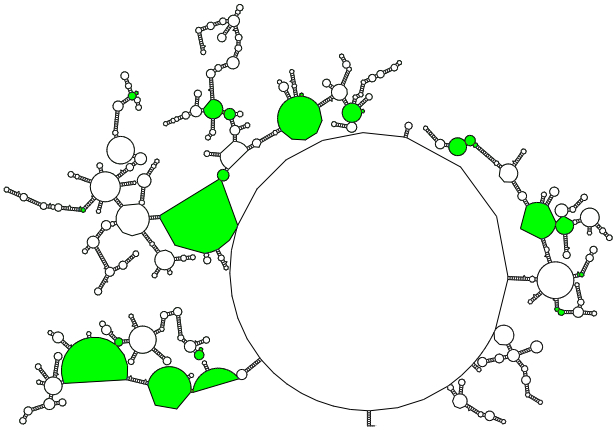


Fig. SI 55: B.subtilis, 23S rRNA

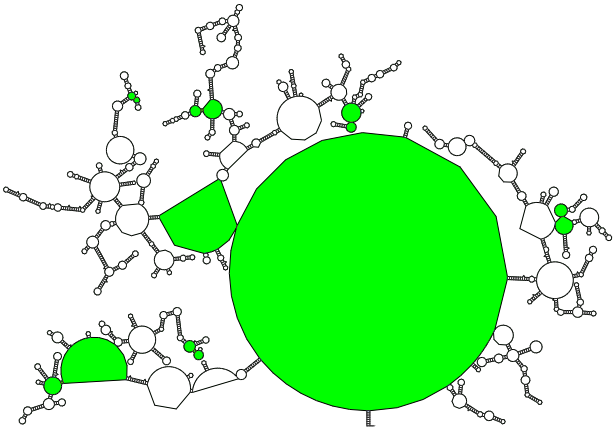


Fig. SI 56: B.subtilis, 23S rRNA

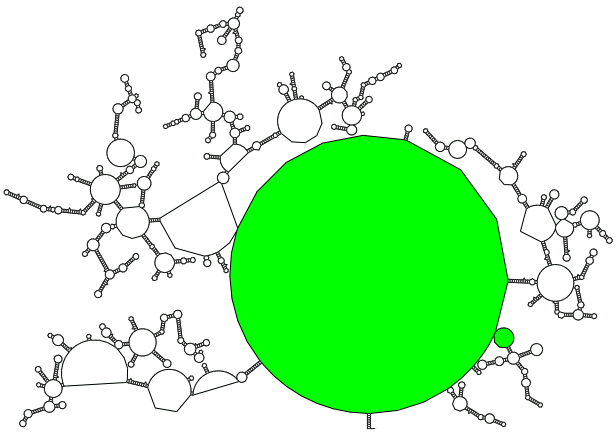


Fig. SI 57: B.subtilis, 23S rRNA

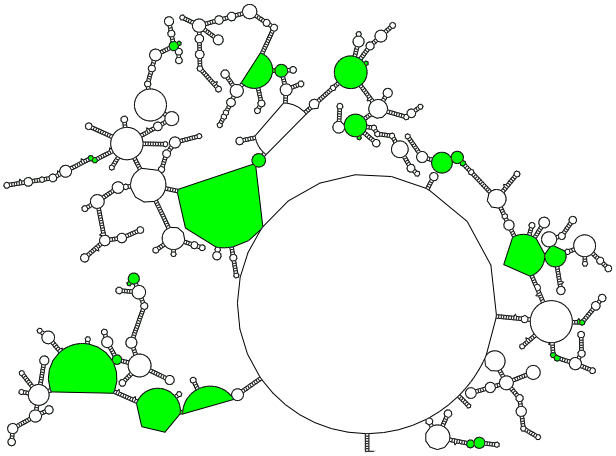


Fig. SI 58: E.coli, 23S rRNA

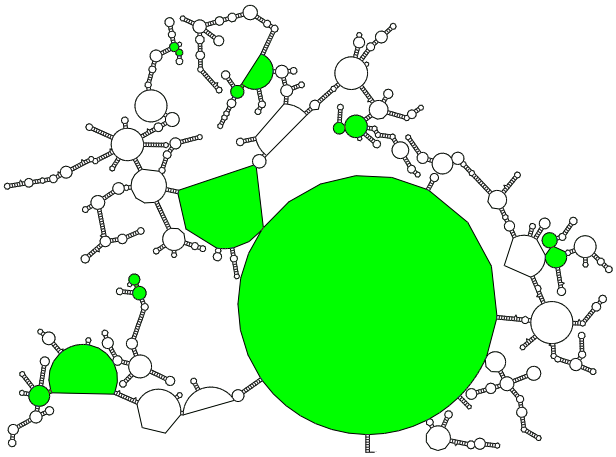


Fig. SI 59: E.coli, 23S rRNA

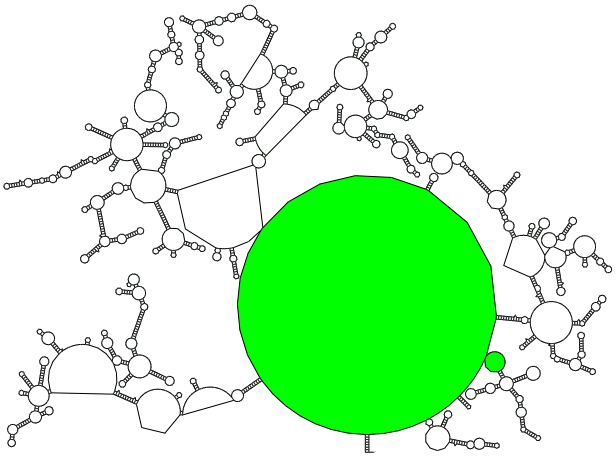


Fig. SI 60: E.coli, 23S rRNA

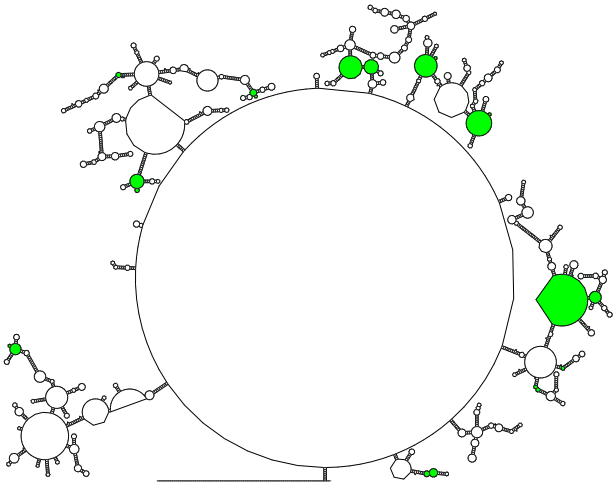


Fig. SI 61: H.pylori, 23S rRNA

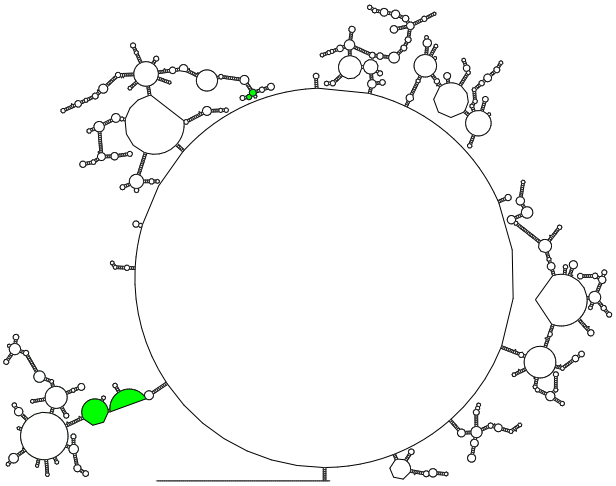


Fig. SI 62: H.pylori, 23S rRNA

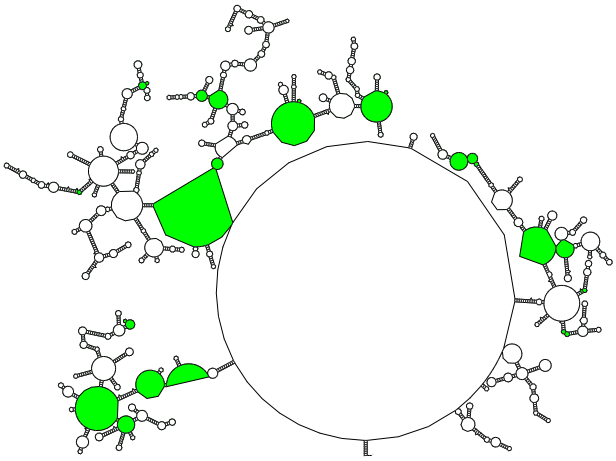


Fig. SI 63: S.aureus, 23S rRNA

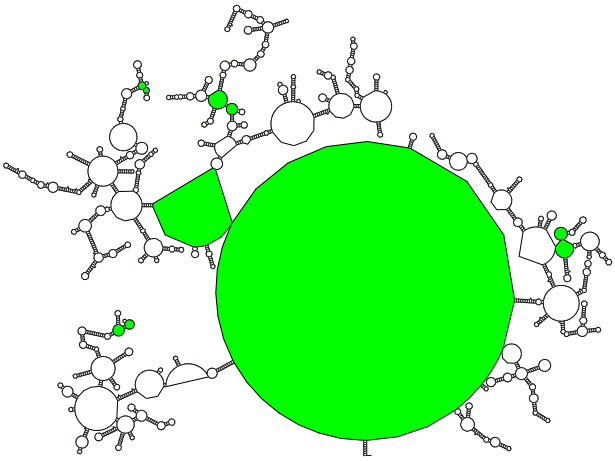


Fig. SI 64: S.aureus, 23S rRNA

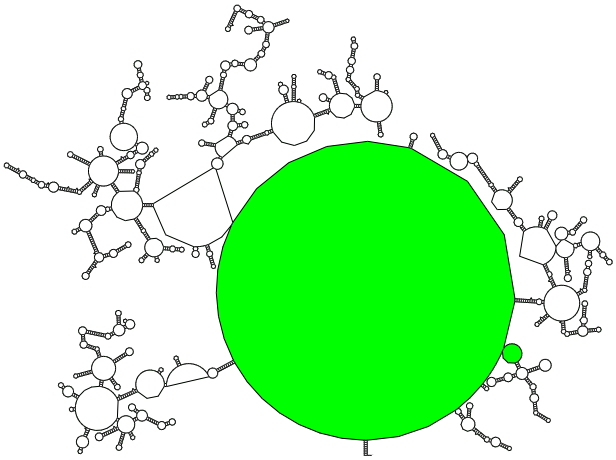


Fig. SI 65: S.aureus, 23S rRNA

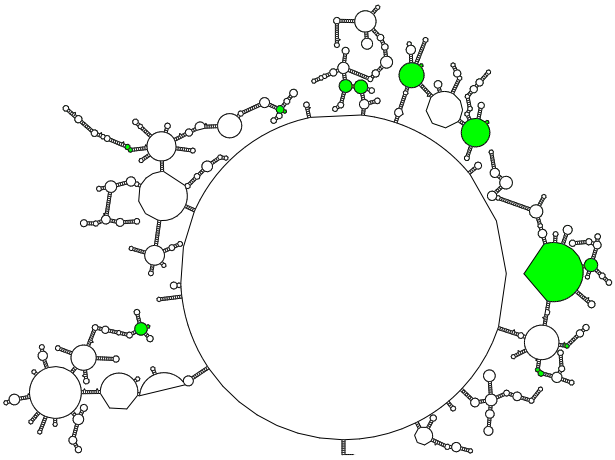


Fig. SI 66: T.thermophilus, 23S rRNA

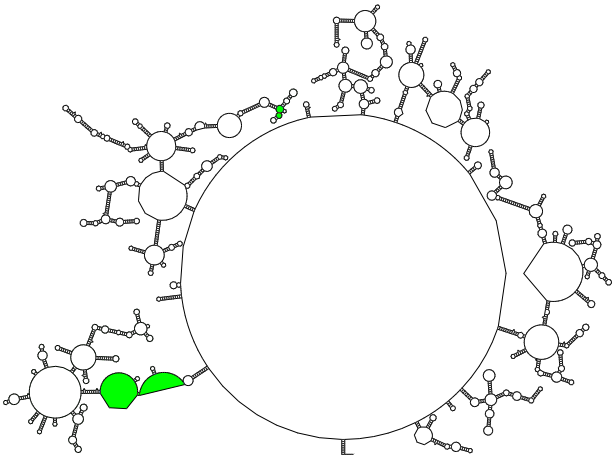


Fig. SI 67: T.thermophilus, 23S rRNA