Forbidden substrings and the connectivity of the Hamming graph of RNA sequences: Partial disconnectivity tests

Maher Mallem, Alain Denise, Yann Ponty

Extended abstract

SeqBIM

Forbidden substrings and the connectivity of the Hamming graph of RNA sequences: partial disconnectivity tests

Maher Mallem[1], Alain Denise[2]*, Yann Ponty[3]*

[1] *Department of Computer Science, ENS Paris-Saclay, Cachan, France*
[2] *LRI and I2BC, Université Paris-Sud / Paris-Saclay, Gif-sur-Yvette, France*
[3] *LIX, École Polytechnique, Palaiseau, France*
***Corresponding authors**: alain.denise@u-psud.fr and yann.ponty@lix.polytechnique.fr

Abstract

RNA structure design methods have grown in complexity to cover an increasing scope of application. Recent approaches combine an initial random generation with a local optimization step, and consider both a user-specified secondary structure and sets of mandatory and forbidden substrings. Although these additional constraints lead to better design results, they may interfere with the local optimization phase. Indeed, forbidden substrings may disrupt the connectivity of their underlying search space, a key property for the success of the local search. A naive connectivity test would explore the whole graph of candidate sequences, leading to an exponential time connectivity test.

In this work, we propose two partial algorithms based on compact graph structures - the De Bruijn graphs and the Aho-Corasick automaton - allowing the detection of disconnectivity in time independent from the length of RNA sequence. Tested on random instances, our tests were able to detect the disconnectivity with sensitivity ranging between 35% and 55%, motivating further research.

Keywords

RNA Design – Forbidden Substrings – De Bruijn graphs – Aho-Corasick automaton

1. Introduction

First introduced in [1], the computational design of RiboNucleic Acids (RNA) design has been studied extensively over the past decades [2] due to its successful application in a variety of biological contexts [3, 4]. Its ultimate goal is the synthesis of molecules to achieve a targeted biological function. In its simplest form, also called **inverse folding** of RNA, the design problem consists in finding a sequence that adopts a given secondary structure as its Minimum Free Energy (MFE) structure, typically computed using polynomial-time dynamic programming [5]. Given the NP-hardness of the problem [6], recent methods [7, 8, 9, 10] tackle the problem heuristically in two phases: First, an initial **seed sequence** is sampled from a distribution that captures a relaxed version of the objective function [11]; Next, the seed is iteratively refined using a **local search strategy** [1], eventually inducing a Boltzmann-Gibbs distribution with respect to the final objective function (*e.g.* the free-energy difference between the sequence MFE structure and its first suboptimal structure).

However, realistic applications of design require additional **sequence constraints**, for instance to avoid undesired interactions within a cellular context. The seed sampling phase can be adapted to avoid a predefined set $\mathcal{F}$ of **forbidden motifs** using formal language constructs [12] or direct dynamic programming [13]. However, to the best of our knowledge, little to no work has been done to assess the **impact of forbidden motifs on the local search**. Indeed, allowing the local search to violate sequence constraints would lead to very few valid candidate sequences, since an overwhelming proportion of the sequences may (and will, from the monkey/typewriter *paradox*) feature some forbidden motif during the local search.

On the other hand, enforcing the avoidance of $\mathcal{F}$ at each step of the local search may disrupt the **search space connectivity**, or equivalently the non-ergodicity of the Markov Chain induced by the sequence space and the moves set of the local search. For instance, while designing an RNA of length n within an alphabet $\Sigma = \{\mathsf{A}, \mathsf{U}\}$ and $\mathcal{F} = \{\mathsf{AU}, \mathsf{UA}\}$, the only two words avoiding $\mathcal{F}$, A^n and U^n, have Hamming distance n. The search space is thus disconnected for any move set inducing changes of bounded Hamming distance $n' < n$. Such a **disconnectivity** prevents the convergence of the local search, *i.e.* it rules out any (probabilistic) guarantee to ultimately discover promising candidates whenever such candidates exist.

In this work, we address the efficient algorithmic detection of disconnected search spaces for a given set $\mathcal{F}$ of forbidden motifs, a given RNA sequence length n and a given moves set. We restrict our attention to k-Hamming move sets, consisting of symmetric moves $s \leftrightarrow s'$ where both s and s' avoid $\mathcal{F}$, and such that Hamming distance $H(s, s') = k$. A *brute-force* solution would generate the whole search space as a graph, and check the existence of a single connected component in a highly impractical $\mathcal{O}(|\Sigma|^n)$ time complexity. Instead, we exploit the highly-structured nature of the problem to propose partial algorithms, based on the De Bruijn graphs and Aho-Corasick automata, whose complexity depend on $\mathcal{F}$ and k, but remain largely independent from n.

2. Definition of the problem

Let Σ be an alphabet, $|\Sigma| \geq 2$, and $n \in \mathbb{N}, n \geq 2$ be a sequence length. Denote by $\mathcal{F} \subset \Sigma^\star$ the set of forbidden motifs, then $\mathcal{L}_{\mathcal{F},n} \subseteq \Sigma^n$ represents the words that do not contain any motif in $\mathcal{F}$. Let $m(\mathcal{F}) \overset{\mathsf{def}}{=} \max_{f \in \mathcal{F}} |f|$, we assume that $n \gg m(\mathcal{F})$.

General problem

Input: Length $n \geq 2$, set $\mathcal{F}$ of forbidden motifs, and neighborhood function $\delta : \mathcal{L}_{\mathcal{F},n} \to \mathcal{L}_{\mathcal{F},n}$
Output: Yes if $G = (\mathcal{L}_{\mathcal{F},n}, \delta)$ is (strongly) connected, No otherwise.

Here we restrict our attention to the k-Hamming neighborhood δ_k for some $k \in [1, n]$, defined for any word $w \in \mathcal{L}_{\mathcal{F},n}$ as $\delta_k(w) = \{w' \in \mathcal{L}_{\mathcal{F}} \mid H(w, w') \leq k\}$ where $H(w, w')$ is the classic Hamming distance between two words w, $w' \in \Sigma^n$.

Since k-Hamming neighborhoods are symmetric, strong connectivity and connectivity are equivalent. The central question, addressed in the following, becomes:

Is the **Hamming graph** $G_{\mathcal{F},n,k} \overset{\mathsf{def}}{=} (\mathcal{L}_{\mathcal{F},n}, \delta_k)$ connected?

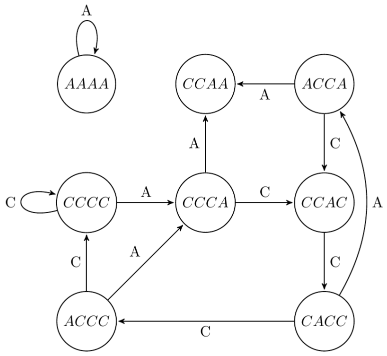


Figure 1. De Bruijn graph $\mathcal{DB}_{\mathcal{F}}$ for $\mathcal{F} = \{\mathsf{ACA}, \mathsf{CAAA}, \mathsf{AAC}\}$ and $\Sigma = \{\mathsf{A}, \mathsf{C}\}$

3. Algorithms

We derive a first partial disconnectivity test from a simple property of De Bruijn graphs. Then using an equivalence relation on the nodes of the De Bruijn graph, we infer a similar partial disconnectivity test on a variant of the Aho-Corasick automaton which is in linear time on the length of the desired sequence.

3.1 Detecting disconnectivity using the De Bruijn graph of $m(\mathcal{F})$-mers

We use variants of the De Bruijn graph [14] to infer the disconnectivity of $G_{\mathcal{F},n,k}$.

Definition 1. *Given a set $\mathcal{F}$ of forbidden motifs, we define:*

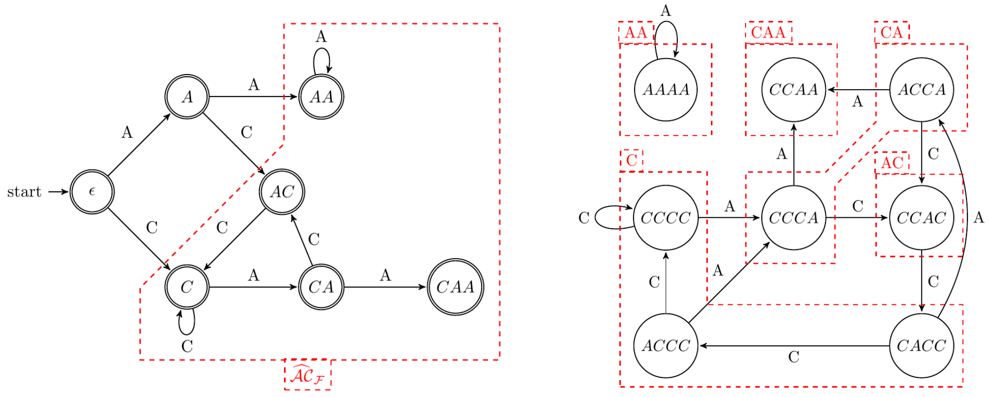
- *The **De Bruijn (di)graph** $\mathcal{DB}_{\mathcal{F}} = (V, E)$ of $\mathcal{F}$, such that $V := \mathcal{L}_{\mathcal{F},m(\mathcal{F})}$, the valid sequences of length $m(\mathcal{F})$, and $E := \left\{(a.w, w.b) \in \mathcal{L}^2_{\mathcal{F},m(\mathcal{F})} \mid a, b \in \Sigma\right\}$;*
- *The **pruned De Bruijn graph** $\mathcal{DB}_{\mathcal{F},n}$, obtained by removing any connected component in $\mathcal{DB}_{\mathcal{F}}$ that cannot generate any word of length n.*

$\mathcal{DB}_{\mathcal{F},n}$ can be built in $\mathcal{O}(|\Sigma|^{m(\mathcal{F})+1})$ time, and detecting unproductive connected components (CC) to build $\mathcal{DB}_{\mathcal{F},n}$ can be done in $\mathcal{O}(|V|)$ time using topological sorting to either detect a cycle ($\rightarrow$ keep CC), or determine n' the length of the longest path ($\rightarrow$ keep CC only if $n' \geq n - m(\mathcal{F}) - 1$).

Remark that $\mathcal{DB}_{\mathcal{F}}$ has $\mathcal{O}(|\Sigma|^{m(\mathcal{F})})$ nodes, and is typically much smaller than the Hamming graph $G_{F,n,k}$ ($\mathcal{O}(|\Sigma|^n)$ nodes), all valid sequences of length n are represented in $\mathcal{DB}_{\mathcal{F}}$ as paths of length $n - m(\mathcal{F})$. For example in Figure 1 the valid sequence CACCAA corresponds to the path CACC → ACCA → CCAA.

Lemma 1. *Upon reading a sequence of letters $a_1.a_2 \ldots a_j$, $j \geq m(\mathcal{F})$ from two distinct nodes $u, v \in \mathcal{DB}_{\mathcal{F}}$ the two paths merge at some index $i \leq m(\mathcal{F})$.*

Intuitively, $\mathcal{DB}_{\mathcal{F}}$ can be seen as an automaton, whose states encode the suffixes of length $m(\mathcal{F})$. Thus, after reading $m(\mathcal{F})$ characters the resulting state is $a_1 \ldots a_{m(\mathcal{F})}$, irrespectively of the starting state, so the paths either merged at index $m(\mathcal{F})$ or before. This means that if we follow two paths in different connected components of $\mathcal{DB}_{\mathcal{F}}$, the sequence of letters must diverge at least once every $m(\mathcal{F})$ steps, which implies an increasing Hamming distance between the corresponding valid words. This



(a) $\mathcal{AC}_{\mathcal{F}}$ **(b)** $\mathcal{DB}_{\mathcal{F}}$ and the equivalence classes

Figure 2. $\widehat{\mathcal{AC}}_{\mathcal{F},n}$ and $\mathcal{DB}_{\mathcal{F},n}$ when $\mathcal{F} = \{\mathsf{ACA}, \mathsf{AAC}, \mathsf{CAAA}\}$ and $\Sigma = \{\mathsf{A}, \mathsf{C}\}$.

holds for any pair of paths in $\mathcal{DB}_{\mathcal{F}}$ generated from different connected components, leading to the following result.

Theorem 2. *$\forall n \geq (k+1) \times m(\mathcal{F}), \mathcal{DB}_{\mathcal{F},n}$ disconnected $\Rightarrow G_{\mathcal{F},n,k}$ disconnected.*

The implication is not an equivalence, as it is possible to build instances where $G_{\mathcal{F},n,k}$ is disconnected while $\mathcal{DB}_{\mathcal{F},n}$ remains connected. It nevertheless suggests a first algorithm for a partial disconnectivity test within $G_{\mathcal{F},n,k}$: Build $\mathcal{DB}_{\mathcal{F},n}$ and report its connectivity. It has overall time complexity in $\mathcal{O}(|\Sigma|^{m(\mathcal{F})})$, *i.e.* no longer exponential in the sequence length n, yet remains exponential in the length of the forbidden substrings.

3.2 Detecting disconnectivity using the Aho-Corasick automaton of $\mathcal{F}$

Next we attempt to exploit the Nerode equivalence, with respect to the suffix language, of some states in $\mathcal{DB}_{\mathcal{F},n}$.

Definition 3. *Define the Aho-Corasick automaton $\mathcal{AC}_{\mathcal{F}}$ as the DFA having states set $Q = \{u$ proper prefix of some $f \in \mathcal{F}\}$, initial state $q_I = \{\varepsilon\}$, and accepting all words ending in Q. Transitions are $\Delta = \Delta_f \uplus \Delta_b$, with:*

- *Δ_f the forward edges: $\{(u, a, u.a) \mid a \in \Sigma \wedge u, u.a \in Q\}$ (i.e. prefix tree of $\mathcal{F}$)*
- *Δ_b the backward edges: $\{(u, a, v) \mid ua \notin Q \wedge v \in Q$ longest suffix of $u.a\}$*

With this definition of $\mathcal{AC}_{\mathcal{F}}$, a word w is accepted iff no $f \in \mathcal{F}$ is a substring of w, *i.e.* $\mathcal{AC}_{\mathcal{F}}$ recognizes the complement language of the usual Aho-Corasick automaton [15]. Moreover, $\mathcal{AC}_{\mathcal{F}}$ can be built in time $\mathcal{O}(|\Sigma| \times |\mathcal{F}| \times m(\mathcal{F}))$.

Definition 4. *We define:*

- *$\widehat{\mathcal{AC}}_{\mathcal{F}}$ from $\mathcal{AC}_{\mathcal{F}}$ by removing states that are no longer visited after $m(\mathcal{F})$ steps;*
- *$\widehat{\mathcal{AC}}_{\mathcal{F},n}$ as the restriction of $\widehat{\mathcal{AC}}_{\mathcal{F}}$ to components producing words of length n.*

$\lvert\Sigma\rvert$	$m(\mathcal{F})$	n	#Samples	$\#G_{\mathcal{F},n,1}$ discon.	%Rec. $\mathcal{DB}_{\mathcal{F},n}$	%Rec. $\widehat{\mathcal{AC}}_{\mathcal{F},n}$
2	5	10	100 000	36 630	49.5	47.1
2	5	11	100 000	35 893	48.2	46.2
3	5	10	10 000	4 395	53.9	49.2
4	3	6	25 000	9 447	37.6	34.3
4	3	7	10 000	3 728	37.9	35.7
4	4	8	4 000	1 904	54.3	50.1

Figure 3. Recall (TP/P) of our disconnectivity tests for various sets of parameters

As illustrated in Figure 2, grouping together nodes in $\mathcal{DB}_{\mathcal{F}}$ having same prefix/suffix overlaps with forbidden substrings, we get exactly $\widehat{\mathcal{AC}}_{\mathcal{F}}$. This equivalence relation and Theorem 2 imply the following:

Theorem 5. $\forall n \geq (k+1) \times m(\mathcal{F})$, *one has*

$$\widehat{\mathcal{AC}}_{\mathcal{F},n} \textit{ disconnected} \Rightarrow \mathcal{DB}_{\mathcal{F},n} \textit{ disconnected} \Rightarrow G_{\mathcal{F},n,k} \textit{ disconnected.}$$

Again, the second implication is only one-way: $\mathcal{DB}_{\mathcal{F},n}$ may be disconnected while $\widehat{\mathcal{AC}}_{\mathcal{F},n}$ remains connected. Still, building $\widehat{\mathcal{AC}}_{\mathcal{F},n}$, and testing its disconnectivity represents an additional partial disconnectivity test for $G_{\mathcal{F},n,k}$. While this variant is expected to detect less cases of disconnectivity, its complexity is significantly better, with the overall construction of $\widehat{\mathcal{AC}}_{\mathcal{F},n}$ now only requiring $\mathcal{O}(\lvert\Sigma\rvert \times \lvert\mathcal{F}\rvert \times m(\mathcal{F}))$ time.

4. Results and Discussion

Both our partial tests were executed on randomly generated sets of forbidden substrings with various parameters. Since the connectivity of the Hamming graph $G_{\mathcal{F},n,k}$ had to be checked on every instance to establish a ground truth, tests could only be conducted with $k = 1$ and small n and $m(\mathcal{F})$ values. The recall (#DetectedDisconnections/#Disconnections, or TP/P) results are given in Figure 3. As expected, the Aho-Corasick-based test always performs slightly worse than the De Bruijn-based one, but not by a large margin ($\sim 5\%$) in our empirical experiments. With a trade-off in accuracy that minimal, the Aho-Corasick-based variant seems to represent a natural first choice in most cases. Recall values range between 35% and 55% for both variants, which is already significant but could probably be improved by exploring subtler relationships between the Aho-Corasick automaton and the Hamming graph.

This preliminary work leaves open several questions of general interest, including:

- What are the shared properties of disconnected instances associated with connected $\widehat{\mathcal{AC}}_{\mathcal{F},n}$? $\mathcal{DB}_{\mathcal{F},n}$?
- Is the problem NP-hard in general?
- How to generalize our constructs to mandatory motifs? To any general automaton generating sequences?
- How to design move sets ensuring connectivity for a given $\mathcal{F}$?

References

[1] Ivo Hofacker, Walter Fontana, Peter Stadler, Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie/Chemical Monthly*, 125(2):167–188, Feb 1994.

[2] Alexander Churkin, Matan Drory Retwitzer, Vladimir Reinharz, Yann Ponty, Jérôme Waldispühl, and Danny Barash. Design of RNAs: comparing programs for inverse RNA folding. *Briefings in Bioinformatics*, 19(2):350–358, 01 2017.

[3] Sven Findeiß, Manja Wachsmuth, Mario Mörl, and Peter F Stadler. Design of transcription regulating riboswitches. In *Methods in enzymology*, volume 550, pages 1–22. Elsevier, 2015.

[4] Ryota Yamagami, Mohammad Kayedkhordeh, David H Mathews, and Philip C Bevilacqua. Design of highly active double-pseudoknotted ribozymes: a combined computational and experimental study. *Nucleic acids research*, 47(1):29–42, 2018.

[5] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9:133–148, 1981.

[6] Édouard Bonnet, Paweł Rzazewski, and Florian Sikora. Designing RNA secondary structures is hard. In *Research in Computational Molecular Biology - 22nd Annual International Conference, RECOMB 2018*, pages 248–250, 2018.

[7] Joseph N Zadeh, Brian R Wolfe, and Niles A Pierce. Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of Computational Chemistry*, 32(3):439–52, 2011.

[8] Matan Drory Retwitzer, Vladimir Reinharz, Yann Ponty, Jérôme Waldispühl, and Danny Barash. incaRNAfbinv: a web server for the fragment-based design of RNA sequences. *Nucleic acids research*, 44(W1):W308–W314, 2016.

[9] Stefan Hammer, Birgit Tschiatschek, Christoph Flamm, Ivo L Hofacker, and Sven Findeiß. RNAblueprint: flexible multiple target nucleic acid sequence design. *Bioinformatics*, 33(18):2850–2858, 04 2017.

[10] Stefan Hammer, Wei Wang, Sebastian Will, and Yann Ponty. Fixed-parameter tractable sampling for RNA design with multiple target structures. *BMC bioinformatics*, 20(1):209, 2019.

[11] Vladimir Reinharz, Yann Ponty, and Jérôme Waldispühl. A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. *Bioinformatics*, 29(13):i308–i315, 2013.

[12] Yu Zhou, Yann Ponty, Stéphane Vialette, Jérôme Waldispühl, Yi Zhang, and Alain Denise. Flexible RNA design under structure and sequence constraints using formal languages. In *ACM-BCB - ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics - 2013*, Bethesda, Washigton DC, United States, September 2013.

[13] Vincent Le Gallic, Alain Denise, and Yann Ponty. Résultats algorithmiques pour le design d'ARN avec contraintes de séquence. In *SeqBio 2015*, pages 26–31, Orsay, France, November 2015.

[14] N. G. De Bruijn. A combinatorial problem. *Proc. Koninklijke Nederlandse Academie van Wetenschappen*, 49:758–764, 1946.

[15] Alfred V. Aho and Margaret J. Corasick. Efficient string matching: An aid to bibliographic search. *Commun. ACM*, 18(6):333–340, June 1975.