Nucleic Acids design targeting integer-valued features: FPT counting and uniform sampling

Yann Ponty, Sebastian Will, Stefan Hammer

Nucleic Acids design targeting integer-valued features: FPT counting and uniform sampling

Yann Ponty [1], Sebastian Will [2], Stefan Hammer [3]

[1] LIX, CNRS & Ecole Polytechnique, Palaiseau, France;
[2] TBI, University of Vienna, Austria;
[3] Bioinformatics, University of Leipzig, Germany.

This article has been accepted for presentation at the 2nd International Workshop on Enumeration Problems and Applications (WEPA 2018).

The computational analysis of RiboNucleic Acids (RNAs) has inspired multiple algorithmic developments, essentially due to the combinatorial nature of the abstract RNA structure representations. While RNA structure prediction has received most of the attention, multiple contributions to Biology and Bioinformatics discuss the rational design of RNAs. Here, we address the problem of **RNA positive design**. It asks for randomly sampling RNAs that satisfy specific constraints and multiple prescribed properties from a controlled (e.g. uniform) probability distribution.

An RNA molecule can be abstracted as a sequence S of length n over the alphabet $\Sigma = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}$. Typically, specific constraints and properties are required for the cellular function of specific RNAs; e.g. the GC-content (number of C and G) must be restricted, or certain sub-words (motifs) must be present or excluded (at arbitrary positions) [14]. Moreover, RNAs may fold into specific structures, materializing hydrogen bonds as a set of **base pairs** $(i, j) \in [1, n]^2$. An RNA can adopt a structure if for each of its base pairs (i, j), the letters S_i and S_j are **compatible**, *i.e.* $\{S_i, S_j\} \in \{\{\mathsf{A}, \mathsf{U}\}, \{\mathsf{G}, \mathsf{C}\}, \{\mathsf{G}, \mathsf{U}\}\}$. The constraints induced by base pair form a compatibility graph $G = (V, E)$, containing a vertex for each position $[1, n]$ and an edge for each base pair. Complex sequence properties, like the GC-content or the free-energy of each target, can be expressed via features. We define **feature functions** $F : \Sigma^* \to \mathbb{N}$, mapping each sequence to a set of integer values. Each feature F is defined additively over a set of **atomic functions** $\{\phi_i\}_i$, each returning a bounded integer value depending on the letters assigned to a specific subset of positions. Moreover, assuming a linear number of ϕ_i, F takes values in $\mathcal{O}(n)$.

Problem statement. Given a compatibility graph $G = (V, E)$ and a set of objective values $f_1^\star, \cdots, f_d^\star$ for the d features $F_1, \cdots, F_d$, return t sequences, drawn uniformly

at random from the sequences S in Σ^n that satisfy Eq. (1) (if such sequences exist at all):

$$\forall (i,j) \in E : S_i \text{ and } S_j \text{ are compatible} \quad \text{and} \quad \forall 1 \leq \ell \leq d : F_\ell(S) = f^\star_\ell. \tag{1}$$

Notably, this problem is a case of uniform sampling from a constraint satisfaction problem with interesting properties; already note that the global constraints $F_\ell(S) = f^\star_\ell$ add complexity beyond typical sampling from a constraint network with local constraints (having constantly bounded arity) [4, 9].

Our approach is ultimately based on counting the admissible sequences. We establish a recursive decomposition of the set of admissible sequences, which we exploit to count and, in turn, generate uniformly at random. This approach relies on dynamic programming, and is akin to the recursive method introduced in the context of enumerative combinatorics [13, 6].

Computational complexity aspects. Earlier work in RNA design sampling [11] focused on a less general version of the problem without feature-based global constraints ($d = 0$); this leaves the compatibility constraints due to G. If G contains any odd cycle, Eq. (1) cannot be satisfied [7]; otherwise, G is bipartite. If G consists of a single component, the number of admissible sequences is exactly twice the number of independent sets in G [10], implying the #P-hardness of counting the number of admissible sequences [8]. While this does not formally imply hardness, the enumeration and uniform random generation problems share a deep connection [12], so this result strongly suggests hardness for our generation problem.

FPT solution based on tree decomposition. In precomputation, we perform cluster tree elimination [3], based on an, arbitrarily oriented, tree-decomposition $T = (V_T, E_T)$ of $G = (V, E)$ [10]. Formulated as a classic dynamic programming scheme, we define the *number of admissible assignments for u under assignment A*, for each node $u \in V_T$, which is identified with a set of positions, and for all (locally admissible) partial assignments $A := p_1 \to b_1, \cdots, p_k \to b_k$ of letters in Σ to the positions u:

$$M[u; A] = \sum_{\substack{\text{partial assignment } A' \text{ of positions } u^\uparrow \\ \text{s.t. } \forall (p,p') \in E : (A \cup A')[p] \text{ comp. w/ } (A \cup A')[p']}} \prod_{(u,v) \in E_T} M[v; (A \cup A')|_v] \tag{2}$$

where $u^\uparrow$ is a shorthand for $u \cap p(u)$, $p(u)$ being the parent of u in T, and $A|_u$ denotes the restriction of an assignment A to u, $A[p]$ is the value assigned to p by A. The total number of admissible sequences is then given by $M[r; \varnothing]$, for r the root of T. Due to the unambiguous decomposition, we directly generate samples based on Eq. (2): From an initial state $(r; \varnothing)$, return $\perp$ if $M[r; \varnothing] = 0$. Otherwise, from any state $(u; A)$, choose an assignment A' with probability $\prod_v M[v; (A \cup A')|_v] / M[u; A]$, and recurse over states $(v; (A \cup A')|_v)$ until the leaves are reached.

The matrix M can be computed in time $\Theta(n \cdot m \cdot |\Sigma|^{w+1})$, with n the RNA length, m the number of input structures, and $w := \max_{u \in T} |u| - 1$ the tree-width of T. Notably, we can decrease the time complexity to $\mathcal{O}(n \cdot \min(w, m) \cdot 2^{w+1} + t \cdot n \cdot \min(w, m))$, including the generation of t sequences, since: i) each of the connected components (CCs) can be treated independently; ii) adjacent positions in G must be assigned to compatible letters, allowing ≤ 2 letters for each position once a first position is assigned in the CC; iii) T can be adapted in such a way that $|u^{\uparrow}| = 1$, while preserving the tree width.

Integrating additive features. A naive approach to capture d integer-valued features would be to modify Eq.(2) to compute $M[u; A, f_1 \cdots f_d]$, the number of assignments to the subtree u, under partial assignment A, such that $F_i = f_i, \forall i \in [1, d]$. Such a **classified DP** algorithm could use explicit convolution products, along with a suitable backtrack to ensure prescribed values for the features, as done in the context of context free languages [5]. This would increase the time complexity to $\mathcal{O}(n^{1+2d} \cdot 2^{w'+1} + m \cdot n^{1+d} \cdot t)$, w' being the width of a tree decomposition for G', a version of G augmented with hyperedges for the set of positions used by atomic functions ϕ.

Alternatively, we used **multidimensional Boltzmann sampling** [2], a technique that relaxes the constraint, induces a suitable distribution and performs a rejection step to target specific feature values. The algorithm first extends (2) to induce a Boltzmann-Gibbs distribution over the feature values, parametrized by weights $w_1, \ldots, w_d$, such that $\mathbb{P}(S; w_1, \ldots, w_d) \propto \prod_\ell w_\ell^{F_\ell(S)}$. The complexity of sampling in this distribution remains mainly the same as the one of Eq. (2) although being based on G', leading to an increased tree width w'. Note that this distribution is uniform within the set of sequences having a given vector of feature values. Rejecting all sequences with wrong feature values thus yields an exact algorithm for our problem. Since typically, induced distributions are **concentrated**, often Gaussian with square-root deviations in $\Theta(\sqrt{n})$, the success rate can be optimized by shifting the sample means to the targeted objectives. For this purpose, one can perform learning by iterative updates of the weights or apply more advanced strategies. For example DFT allows to compute $M[r; \varnothing, f_1^\star, \cdots, f_d^\star]$ in time $\mathcal{O}(dn \log n)$, which can even test the existence of a solution. Finally, by tolerating a positive target deviation $\epsilon > 1/\sqrt{n}$, one expects only a constant number of rejections per sample; even with zero tolerance, this number goes up to (only) $\Theta(n^{d/2})$.

Constraining sequence motifs. Forbidding a set of subwords $\mathcal{W}$ of length $\leq k$ could be achieved by adding k-ary constraints for each k successive sequence positions to the binary complementarity constraints to the objective (1), potentially increasing the tree width. Alternatively, we could adapt a strategy based on building an Aho-Corasick automaton [1] for the sequences, which can be transformed into a deterministic finite automaton (DFA) $\mathcal{M}$ for the language $\mathcal{W}$, as done for the single target design [14]. Instead of considering assignments to nucleotides in Σ, we now

consider assignments to $Q \times \Sigma$, where Q are the states $\mathcal{M}$. We then restrict the value of any pair of consecutive positions to valid transitions of $\mathcal{M}$ and forbid the acceptance states of $\mathcal{M}$ in the assignments (at each position). This modification results in a complexity of $\mathcal{O}(n \cdot m \cdot (|\Sigma| \cdot |Q|)^{w''+1})$, where w'' is the tree width taking the additional dependencies (between successive positions) into account. Again, as previously done [14], a similar strategy would let us *enforce* subwords at arbitrary positions.

Open Questions Various questions remain open, among them: the detailed complexity of generation, the convergence to the *concentrated* distribution and sufficient conditions, and how to (better) ensure uniformity within a range of values? Moreover: is there a maximum tree width w of bi- or k-secondary structures? How does w change due to adding a Hamiltonian path? Finally, how to include negative design aspects? How is w affected by typical negative design constraints (e.g. forbidding all small stable off-target hairpins)?

References

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

[2] Olivier Bodini and Yann Ponty. Multi-dimensional Boltzmann sampling of languages. In *Proceedings of the 21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms (AofA'10)*, pages 49–64. DMTCS Proceedings, 2010.

[3] Rina Dechter. *Reasoning with Probabilistic and Deterministic Graphical Models: Exact Algorithms*. Morgan & Claypool, 2013.

[4] Rina Dechter, Kalev Kask, Eyal Bin, and Roy Emek. Generating random solutions for constraint satisfaction problems. In *Eighteenth National Conference on Artificial Intelligence*, pages 15–21, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[5] Alain Denise, Yann Ponty, and Michel Termier. Controlled non-uniform random generation of decomposable structures. *Theoretical Computer Science*, 411(40-42):3527–3552, 2010.

[6] Philippe Flajolet, Paul Zimmermann, and Bernard Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132(1):1 – 35, 1994.

[7] C. Flamm, I. L Hofacker, S. Maurer-Stroh, P. F Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA (New York, N.Y.)*, 7:254–265, Feb 2001.

[8] Qi Ge and Daniel Štefankovič. A graph polynomial for independent sets of bipartite graphs. *Combinatorics, Probability and Computing*, 21(05):695–714, 2012.

[9] Vibhav Gogate and Rina Dechter. A new algorithm for sampling csp solutions uniformly at random. In Frédéric Benhamou, editor, *Principles and Practice of Constraint Programming - CP 2006*, pages 711–715, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[10] Stefan Hammer, Yann Ponty, Wei Wang, and Sebastian Will. Fixed-Parameter Tractable Sampling for RNA Design with Multiple Target Structures. In *RECOMB 2018 – 22nd Annual International Conference on Research in Computational Molecular Biology*, pages 256–258, Paris, France, April 2018. Extended version available at https://hal.inria.fr/hal-01631277/file/RNARedPrint-Bioinfo2018.pdf.

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

[12] Mark R. Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169 – 188, 1986.

[13] Herbert S. Wilf. A unified setting for sequencing, ranking, and selection algorithms for combinatorial objects. *Advances in Mathematics*, 24(2):281 – 291, 1977.

[14] 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 Jing Gao, editor, *ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics. ACM-BCB 2013, Washington, DC, USA, September 22-25, 2013*, page 229. ACM, 2013.