A Combinatorial Framework for Designing (Pseudoknotted) RNA Algorithms

Yann Ponty, Cédric Saule

A Combinatorial Framework for Designing (Pseudoknotted) RNA Algorithms

Yann Ponty[1⋆] and Cédric Saule[2,3]

[1] LIX, École Polytechnique/CNRS/INRIA AMIB, France
yann.ponty@lix.polytechnique.fr
[2] LRI, Université Paris-Sud/XI/INRIA AMIB, France
[3] Institute for Research in Immunology and Cancer, Montreal, Quebec, Canada
saule@lri.fr

Abstract. We extend an hypergraph representation, introduced by Finkelstein and Roytberg, to unify dynamic programming algorithms in the context of RNA folding with pseudoknots. Classic applications of RNA dynamic programming (Energy minimization, partition function, base-pair probabilities…) are reformulated within this framework, giving rise to very simple algorithms. This reformulation allows one to conceptually detach the conformation space/energy model – captured by the hypergraph model – from the specific application, assuming unambiguity of the decomposition. To ensure the latter property, we propose a new combinatorial methodology based on generating functions. We extend the set of generic applications by proposing an exact algorithm for extracting generalized moments in weighted distribution, generalizing a prior contribution by Miklos and al. Finally, we illustrate our full-fledged programme on three exemplary conformation spaces (secondary structures, Akutsu's simple type pseudoknots and kissing hairpins). This readily gives sets of algorithms that are either novel or have complexity comparable to classic implementations for minimization and Boltzmann ensemble applications of dynamic programming.

Key words: RNA folding, Pseudoknots, Boltzmann Ensemble, Hypergraphs, Dynamic Programming

1 Introduction

Motivation. Over the past decades biology as a field has become increasingly aware of the importance and diversity of roles played by ribonucleic acids (RNA). In addition to playing house-keeping parts, as initially contemplated by the proteo-centric view of cellular processes, RNA is now accepted as a major player of gene regulation mechanisms. For instance silencing activity (miRNAs, siRNAs) or multi-stable cis-regulatory elements (riboswitches) are currently the subject of many research. Furthermore a recent genome-wide experiment has revealed that a large portion of the human genome was subject to transcription into RNA. While it is unlikely for all these transcripts to be functional as RNAs, novel classes and roles are currently under investigation. Most of the functional roles played by RNA require the RNA to adopt a specific structure to make an interaction possible, hide/exhibit an active site or allow for a catalytic action (Ribozymes). Being able to understand and simulate how RNA folds is therefore a crucial step toward understanding its function.

Ab initio secondary structure prediction. Initial algorithmic methods for the ab-initio prediction of RNA folding considered a coarse-grain conformation space, the secondary structure, where each conformation is defined as a non-crossing subset of admissible base-pairs. This led Nussinov and Jacobson [39] to design a $\Theta(n^3)$ dynamic-programming (DP) algorithm for the base-pair maximization problem. Building on a nearest neighbor free-energy model proposed by Tinoco *et al* [51] and extended by the Turner group, Zuker and Stiegler [56] created MFOLD, a $\Theta(n^3)$ algorithm for minimizing the free-energy (MFE folding), later shown to predict correctly ~73% of base-pairs on a benchmark of RNAs of length < 700 nucleotides [34]. An independent implementation of the algorithm is proposed within the popular VIENNARNA package maintained by Hofacker [22]. Probabilistic alternatives (SFOLD [11], CONTRAFOLD [14] and CENTROIDFOLD [20]) have also recently been proposed with substantial improvement, relying on a

⋆ To whom correspondence should be addressed

dynamic programming scheme similar to that of MFOLD to traverse the conformation space in polynomial time coupled with some postprocessing steps.

Ensemble approaches. Since the seminal work of McCaskill [35], the concept of Boltzmann equilibrium has been used to embrace the diversity of folding accessible to an RNA sequence. He showed that the partition function of an RNA – a weighted sum over the set of all compatible structures – could be computed through a simple transposition of the DP scheme used for MFE folding. Coupled with a variant of the inside/outside algorithm, this led to an exact computation of base-pairs probabilities in the Boltzmann-weighted ensemble. This opened the door for more robust predictions, e.g. for RNAs whose MFE folding is an outlier. This intuition was later validated by Mathews [33] who showed that the Boltzmann probability correlated well with the actual presence of base-pairs in experimentally-determined structures. Ding *et al* [11] pushed this paradigm shift a step further by clustering sets of structures sampled within the Boltzmann distribution and computing a consensus, improving on the positive-predictive-value (PPV) of existing algorithms. This ensemble view naturally spread toward other applications of DP in Bioinformatics (sequence alignement [38], simultaneous alignment and folding [21], 3D structural alignement [15]), and is increasingly becoming a part of the *algorithmic toolbox* of bioinformaticians.

Pseudoknotted conformations. Although substantially successful in their task, secondary structure prediction algorithms were intrinsically limited in by their inability to explore conformations featuring crossing base-pairs. Such motifs, called pseudoknots, were initially excluded from the conformation space based on the rationale that their participation to the free-energy would remain limited. Furthermore, the adjunction of all possible pseudoknots was shown to turn MFE folding into an NP-complete problem even in a simple nearest-neighbor model [1, 30]. However such conformations do naturally occur, and can be essential to functional mechanisms such as -1-frameshift recoding events [4] or the formation of tertiary motifs [40]. Therefore many exact DP approaches [45, 30, 13, 42, 6–8, 7, 23, 50, 44] have been proposed over the years to extract the MFE structure within restricted – polynomially solvable – classes of pseudoknots. However most of these approaches (with the notable exceptions of [13, 6, 44]) were based on ambiguous DP schemes, leading them to consider certain structures multiple times. While such an unambiguity would not be worrisome in the context of energy minimization, it prevents a direct transposition of these algorithms to ensemble applications (partition function, base-pair probabilities) by heavily biasing – for no biologically valid reason – derived estimates.

Unambiguous decompositions. This lack of focus on unambiguity in the design of RNA (pseudoknotted) DP algorithms can be explained by two main reasons. Firstly certain conformation spaces may not admit unambiguous schemes. Indeed it has been shown by Condon *et al* [9] that many PK conformational spaces can be modeled as a formal language, while Flajolet [18] had shown, using a combinatorial argument, that certain simple context-free languages are inherently ambiguous, i.e. not generated by any unambiguous context-free grammar. A second explanation is more historical: DP algorithms designers were initially focused on optimization problems, and considered the DP equation, not the decomposition of the search space, as the central object of their contributions. Indeed in the optimization perspective, it is not mandatory for the conformation space to be completely (e.g. sparsification) or unambiguously (e.g. multiply occurring best structure) generated. As decompositions grow more and more complex to capture more complex energy models and topological limitations, these two key properties are becoming increasingly hard to ascertain at the level of DP equations. Consequently there is a need for more rational framework to facilitate the design of conformational spaces.

Combinatorial dynamic programming. Over the last century, enumerative combinatorics as a field has been focusing on providing elegant decompositions for all sorts of objects. Our proposal is to adopt a similar discipline in the design of DP decompositions, the only task worthy of human attention to our opinion, and will eventually lead to an automated procedure for the actual production of codes/algorithms. To that purpose we chose to build on and revisit an hypergraph analogy proposed by Finkelstein *et al* [16] as a unifying framework for RNA folding and other applications of DP in Bioinformatics, which we generalize into combinatorial classes amenable to analysis using generating functions.

Related work. The two main frameworks offering abstracts view over Dynamic Programming are Lefebvre's multi-tape attributed grammars [26] and Giegerich's Algebraic Dynamic Programming (ADP) [19], respectively building on multitape-attributed grammars and context-free grammars. Although very elegant and mature in their implementations, they suffer from limitations in expressivity that are intrinsic to their underlying formalisms. For instance, ADP has to resort to an explicit manipulation of indices

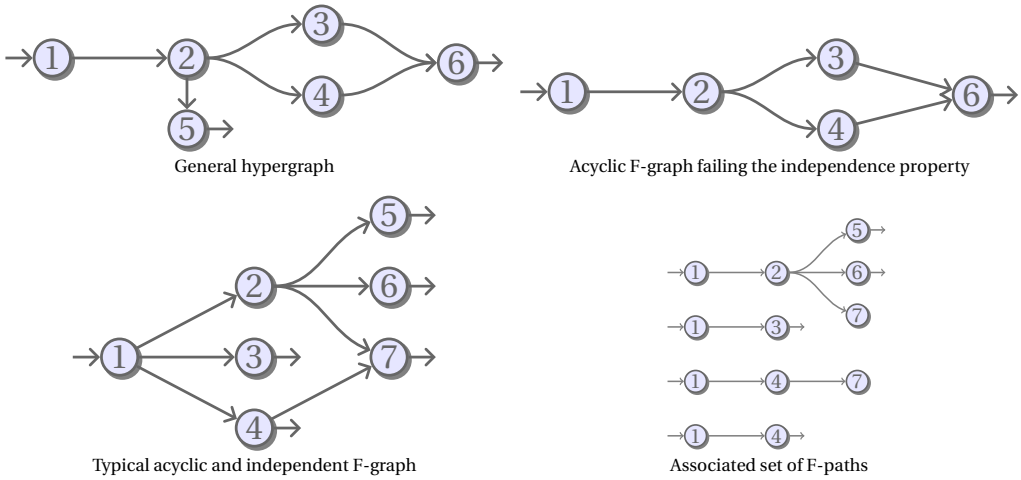

Fig. 1. Illustration of F-Graphs, F-Paths and Independence property. Straight lines indicate classic arcs, and bent lines indicate hyperarcs.

in order to achieve competitive complexities for canonical pseudoknots [42], while Lefebvre's multi-tape grammars [27] require increased complexity to capture pseudoknots. Another formal description of pseudoknotted search spaces is M. Möhl's *split-types* [37], which focuses on how non-contiguous portions are combined, providing a very compact description for pseudoknotted conformation spaces. Compared to these abstract representations, the hypergraph formalism achieves a greater expressivity by: i) Implementing an unordered product; ii) Allowing explicit manipulation of indices; iii) Allowing additional information to be stored within nodes (Remember that context-free grammars allow for a finite number of non-terminals). For instance, polynomial hypergraphs could be proposed for counting homogeneous alignments [25] whereas these objects cannot be generated by any context-free grammar [5] and will not be expressed strictly within the alternative frameworks. This improved expressivity comes at a price since the manual manipulation of indices is error-prone, as pointed accurately by Giegerich et al, so one may want to think of our proposal as more of a byte code, possibly produced from a higher-level source code (ADP, split-types...).

Outline. In Section 2, we briefly remind some basic definitions related to forward directed hypergraphs. In Section 3, we remind and propose dynamic programming algorithms for generic problems on F-graphs. Then in Section 4, we illustrate our programme by proposing and proving unambiguous decompositions for three space of conformations: Classic secondary structures in the Turner energy model [32], (weighted) base-pair maximisation version of Akutsu's simple-type pseudoknots [1] and fully-recursive kissing hairpins (Unambiguous restriction of Chen *et al* [8]). We also describe a simplified proof strategy based on generating functions to prove the correctness of a given decomposition. Section 5 enriches the scope of applications of our framework by proposing a general algorithm for extracting the moments of additive features (free-energy, base-pairs, helices...) in a weighted distribution (generalizing a previous contribution by Miklos *et al* [36]). Finally Section 6 concludes with some remarks and possible extensions and improvements.

2 Notations and key notions

Let us first remind that a directed hypergraph generalizes the notion of directed graph by allowing any number of vertices as origin(**tail**) and destination (**head**) for each (hyper)-arcs. We will be focusing here on Forward-Hypergraphs, or **F-graphs**, which restrict the tail of their arcs to a single vertex.

Formally, let V be a set of vertices, an **F-arc** $e = (t(e) \rightarrow \mathbf{h}(e)) \in V \times \mathscr{P}(V)$, connects a single tail vertex $t(e) \in V$ to an ordered list of vertices $\mathbf{h}(e) \subseteq V$. An **F-graph** $\mathscr{H} = (V, E)$ is characterized by a set of vertices V and a set of F-arcs E. Denote by $\mathbf{c}_n$ the children of a node in a tree, then an **F-path** of $\mathscr{H} = (V, E)$ is a

tree $\mathscr{T} = (V' \subseteq V, E')$ such that, for any node $n \in V'$, $(v_n \to \mathbf{c}_n) \in E$. For the sake of simplicity, we may omit the implicit V' and identify an F-path to its set of edges E'.

An **F-derivation** from a vertex $s \in V$ can be recursively defined as either $\langle s, \varnothing \rangle$ if $(s \to \varnothing) \in E$, or $\langle s, D_1 \dots D_{|\mathbf{t}|} \rangle$ if $(s \to \mathbf{t}) \in E$, $\mathbf{t} = \{t_1, t_2, \dots, t_{|t|}\}$, and each D_i is an F-derivation starting from t_i. An F-graph is **acyclic** if and only if any vertex $s \in V$ is present only once (as a root) in any derivations starting from s. Moreover it is **independent** if and only if any vertex $s \in V$ is reached at most once in any derivation, regardless of its root.

A **weighted F-graph** is a triplet (V, E, π) such that (V, E) is an F-graph and $\pi : E \to \mathbb{R}^+$ is a weight function that associates a weight to each F-arc. Finally, an **oriented F-graph** is a quadruplet (v_0, V, E, π) such that (V, E, π) is a weighted independent F-graph, and $v_0 \in V$ is a distinguished initial vertex.

Remark 1: Notice that our definition of F-arcs and F-paths implicitly defines **terminal vertices**, since any leaf l in a F-path has no child and our definition of F-paths therefore requires $l \to \varnothing$ to be an F-arc of $\mathscr{H}$.

Remark 2: Under the independence property, the derivations starting from any node $s \in V$ are trees, and are therefore in bijection with F-paths originating from the same vertex.

3 Generic problems and algorithms for F-paths in F-graphs

In the following, terminal cases will very seldom appear explicitly, but will rather be captured by the limit cases of products $\prod_{u \in \varnothing} f(u) = 1$ and sums $\sum_{u \in \varnothing} f(u) = 0$, $k \in \mathbb{R}$.

Generating and counting F-paths in oriented F-graphs [55] Let $\mathscr{H} = (v_0, V, E, \pi)$ be an oriented F-graph, we address the problem of generating the set $\mathscr{P}_{v_0}$ of F-paths obtained starting from v_0.

From the tree-like definition of F-paths and our remark on terminal vertices, we know that any F-path starting from a vertex s can either be a leaf, provided that there exists an F-arc $s \to \varnothing$, or an internal node. In the latter case, any F-paths is composed of auxiliary paths, generated from the vertices in the head of some F-edge having s as tail. Remark that our definition of F-paths requires each vertex from V to appear at most once in any F-path, a fact that is ensured here by the acyclicity of $\mathscr{H}$. Therefore we can recursively define the set of $\mathscr{P}_s$ of F-paths starting from a root node s as

$$\mathscr{P}_s = \left\{ \begin{array}{c} \{(s,\varnothing)\} \text{ If } (s,\varnothing) \in E \\ \varnothing \quad \text{Otherwise} \end{array} \right\} \cup \bigcup_{(s \to \mathbf{t}) \in E} \left(\{s\} \times \prod_{u \in \mathbf{t}} \mathscr{P}_u \right), \quad \forall s \in V. \tag{1}$$

Since E is a set, the candidate heads for a given tail s are distinct and the unions in the above equations are disjoint. Furthermore, the products are Cartesian, so we can directly transpose the recurrence above over the cardinalities $n_s = |\mathscr{P}_s|$ and obtain

$$n_s = \sum_{(s \to \mathbf{t}) \in E} \prod_{u \in \mathbf{t}} n_u, \quad \forall s \in V. \tag{2}$$

This immediately yields a $\Theta(|V| + |E| + \sum_{e \in E} |\mathbf{h}(e)|)/\Theta(|V|)$ time/memory dynamic programming algorithm for counting F-paths.

Minimal score F-path Let us consider an **additive scoring scheme** based on weights, and accordingly define the **score** of an F-path p to be $\alpha(p) = \sum_{e \in E} \pi(e)$. We address here the problem of finding an F-path p_0 having minimal score or more formally some $p_0 \in \mathscr{P}_{v_0}$ such that $\forall p \in \mathscr{P}_{v_0},\ p \neq p_0 \Rightarrow \alpha(p) \geq \alpha(p_0)$. From the independence of siblings and the strict additivity of the score, we know that the path minimization problem has optimal substructure, i. e. any optimal solution is composed of optimal solutions for its subproblems. Consequently, the **minimal score** m_s of a path starting from a root node $s \in V$ is given by

$$m_s = \min_{e=(s \to \mathbf{t}) \in E} \left(\pi(e) + \sum_{u \in \mathbf{t}} m_u \right), \quad \forall s \in V. \tag{3}$$

A classic backtrack procedure can then be used to reconstruct the F-path instance $p_s^{\min}$ starting from $s \in V$ and having minimal score. Alternatively, the previous recurrence can be modified as follows

$$p_s^{\min} = \underset{\substack{p' = \bigcup_{s' \in \mathbf{t}} p_{s'}^{\min} \\ \text{s.t. } (s \to \mathbf{t}) \in E}}{\operatorname{argmin}} \alpha\left(\{(s \to \mathbf{t})\} \cup p'\right), \quad \forall s \in V, \tag{4}$$

giving a $\Theta(|V| + |E| + \sum_{e \in E} |\mathbf{h}(e)|)/\Theta(|V|)$ time/memory DP algorithm for the minimal weighted F-path.

Weighted count and weighted random generation [10] Let us **extend multiplicatively on paths** our weight function, defining the **weight of any F-path** p to be $\pi(p) = \prod_{e \in p} \pi(e)$. Then a small modification of Equation 2 gives a recurrence for computing the cumulated weight, or **weighted count** w_s of F-paths starting from a given vertex s:

$$w_s = \sum_{p' \in \mathscr{P}_s} \pi(p') = \sum_{e=(s \to \mathbf{h}(e)) \in E} \pi(e) \cdot \prod_{s' \in \mathbf{h}(e)} w_{s'}, \quad \forall s \in V \tag{5}$$

Provided that the weights are positive, this defines a **weighted probability distribution** over F-paths, which assigns to each path $p \in \mathscr{P}_{v_0}$ a probability

$$\mathbb{P}(p \mid \pi) = \frac{\pi(p)}{\sum_{p' \in \mathscr{P}_{v_0}} \pi(p')} \equiv \frac{\pi(p)}{w_{v_0}}. \tag{6}$$

From the precomputed values w_s, one can perform a **weighted random generation** to draw at random a set of k F-paths from v_0 according to a weighted distribution. Starting from any vertex s, the algorithm chooses at each step an F-arc $e = (s \to \mathbf{h}(e))$ with probability

$$p_{s,e} = \frac{\pi(e) \cdot \prod_{s' \in \mathbf{h}(e)} w_{s'}}{w_s},$$

and proceeds to the recursive generation of auxiliary paths from each vertex in $\mathbf{h}(e)$. A simple induction argument shows that any F-path is then generated with respect to the probability distribution of Equation 6. The weighted count recurrence is computed by a $\Theta(|V| + |E| + \sum_{e \in E} |\mathbf{h}(e)|)/\Theta(|V|)$ time/memory algorithm, and each path p is generated in $\Theta(|p| + \sum_{e \in p} |\mathbf{h}(e)|)/\Theta(|p|)$ time/memory.

Remark 3: This worst-case complexity can be improved using additional information on the structure of the F-graph. For instance, when both the height and maximal degree of a vertex are bounded by some constant n, Boustrophedon search [17, 41] can be used to decrease the worst-case complexity of each generation from $\Theta(n^2)$ to $\mathscr{O}(n \log n)$.

Arc traversal probabilities Using the same probability distribution, a natural problem is to compute the probability p_e of an F-arc $e \in E$ being in a random F-path. To that purpose one can use the classic *inside/outside* algorithm, which can be rephrased as an F-graphs traversal.

Let us first point out that the probability p_e is related to the cumulated weight of all F-paths featuring an edge $e = (t(e) \to \mathbf{h}(e))$ through

$$p_e = \frac{\sum_{\substack{p \in \mathscr{P}_{v_0} \\ \text{s.t. } e \in p}} \pi(p)}{\sum_{p' \in \mathscr{P}_{v_0}} \pi(p')} \equiv \frac{\sum_{\substack{p \in \mathscr{P}_{v_0} \\ \text{s.t. } e \in p}} \pi(p)}{w_{v_0}}. \tag{7}$$

From the independence of $\mathscr{H}$, we know that each vertex appears at most once in any given F-path, and consequently any F-path traversing e can therefore be **unambiguously** decomposed into: i) An **e-outside tree**, i.e. a derivation from v_0 whose leaves are either terminal or $t(e)$, and which features exactly one occurrence of $t(e)$; ii) A **support edge** $e = (t(e) \to \mathbf{h}(e))$; iii) An **e-inside tree**, i.e. a set of F-paths issued from $\mathbf{h}(e)$.

The unambiguity of the decomposition, along with the independence of i) and iii), translates into

$$\sum_{\substack{p \in \mathscr{P}_{v_0} \\ \text{s.t. } e \in p}} \pi(p) = b_{t(e)} \cdot \pi(e) \cdot \prod_{s' \in \mathbf{h}(e)} w_{s'} \tag{8}$$

where b_s is the cumulated weight of all outside trees leaving $s \in V$ underived. Finally it can be shown that the cumulated weight b_s over all e-outside trees obey the following simple recurrence

$$b_s = \mathbf{1}_{s=q_0} + \sum_{\substack{e' \in E \\ \text{s. t. } s \in \mathbf{h}(e')}} \pi(e') \cdot b_{t(e')} \cdot \prod_{\substack{s' \in \mathbf{h}(e') \\ s' \neq s}} w_{s'}, \quad \forall s \in V \tag{9}$$

which can computed in $O(|V| + |E| + \sum_{e \in E} |\mathbf{h}(e)|^2)/\Theta(|V|)$ time/memory. The probability of traversing p_e in a random F-path can finally be computed through the formula

$$p_e = \frac{b_{t(e)} \cdot \prod_{s' \in \mathbf{h}(e)} w_{s'}}{w_{v_0}}, \quad \forall e \in E. \tag{10}$$

4 F-graphs reformulation of (Pseudoknotted) RNA conformation spaces

From the previous section, we know that very simple algorithms exist for weighted optimization and enumeration problems over the F-paths of an F-graph. Let us now consider MFE folding-related problems over an arbitrary **conformation space** D for a sequence ω, under an energy model $E : D \to \mathbb{R}$ and assume that there exists: **C1.** An F-graph $\mathscr{H}$ whose F-paths $\mathscr{P}$ are in bijection with the conformation space D; **C2.** A weight function π such that the (additive) score of any F-path coincides with the energy of its corresponding conformation.

Under such conditions, it can be remarked that the **minimal score** algorithm (Equation 3) exactly computes the **Minimal Free-Energy** $MFE = \min_{s \in D} E_s$. Furthermore, the **Weighted Count** (Equation 5), applied to a weight function $\pi'(e) = e^{-\pi(e)/RT}$, computes the **Partition Function** $\mathscr{Z} = \sum_{s \in D} e^{-E_s/RT}$. Other quantities of interest for RNA folding can also be derived, as summarized in Tables 1 and 2.

4.1 Foreword: Shortening correctness proofs through generating functions

Our main challenge is to find an hypergraph/weight such that the energy function can be expressed in an additive fashion. Focusing first on Condition **C1**, one remarks that finding a function $\psi : \mathscr{P} \to D$ which maps F-Paths to elements of the conformation space is not challenging, as it essentially amounts to figuring out which derivation creates which base-pairs. Condition **C1** is then traditionally broken into two parts: an **unambiguity** condition which requires distinct elements in $\mathscr{P}$ to give rise to distinct elements within D, i.e. ψ should be injective; a **completeness** condition which requires each element in S to have at least one pre-image, i.e. ψ should be surjective.

Since these notions are intimately related to the semantics associated with the F-paths, they cannot be tackled in an automated way at the hypergraph level[4]. Therefore correctness proofs will usually require user-assigned semantics coupled with custom arguments, a task that may become challenging and/or tedious for complex decompositions. In order to simplify the validation and therefore the design of new conformation spaces, we propose a simplified proof technique based on generating functions.

Indeed, instead of specializing the hypergraph for each and every input sequence, one can delegate to the weight function the responsibility of weeding out conformations, e.g. by assigning them $+\infty$ energetic contributions within MFE folding. Therefore each class of conformations can be seen as a family of conformation space $\{D_n\}_{n \geq 0}$ (secondary structures, simple type pseudoknots…), to which one associates a family of hypergraphs $\{\mathscr{H}_n\}_{n \geq 0}$, a **decomposition**, both indexed by the length n of the sequence.

Let us remind that generating functions are formal power series that can be used to store various information. For instance the counting generating function for the conformation space family $\mathscr{D}$ can be defined as $S_{\mathscr{D}}(z) = \sum_{n \geq 0} |D_n| \cdot z^n$ where z is a formal complex variable devoid of intuitive meaning. Furthermore let $\mathscr{P}_n$ be the set of F-Paths associated with $\mathscr{H}_n$, then the counting generating function of the decomposition can be defined as $S_{\mathscr{H}}(z) = \sum_{n \geq 0} |\mathscr{P}_n| \cdot z^n$. Then the formal identity $S_{\mathscr{D}}(z) = S_{\mathscr{H}}(z)$ implies that $|D_n| = |\mathscr{P}_n|, \forall n \geq 0$. It follows from basic set theory that unambiguity/injectivity (resp. completeness/surjectivity) of ψ, in addition to the identity of generating functions, is in itself sufficient to prove the bijectivity of ψ. Since reference generating functions are now available for many conformation space families [47], this practically halves the burden of designing a proof.

[4] Algebraic Dynamic Programming partially addresses this issue, and the interested reader is referred to an early contribution by Reeder *et al* [43].

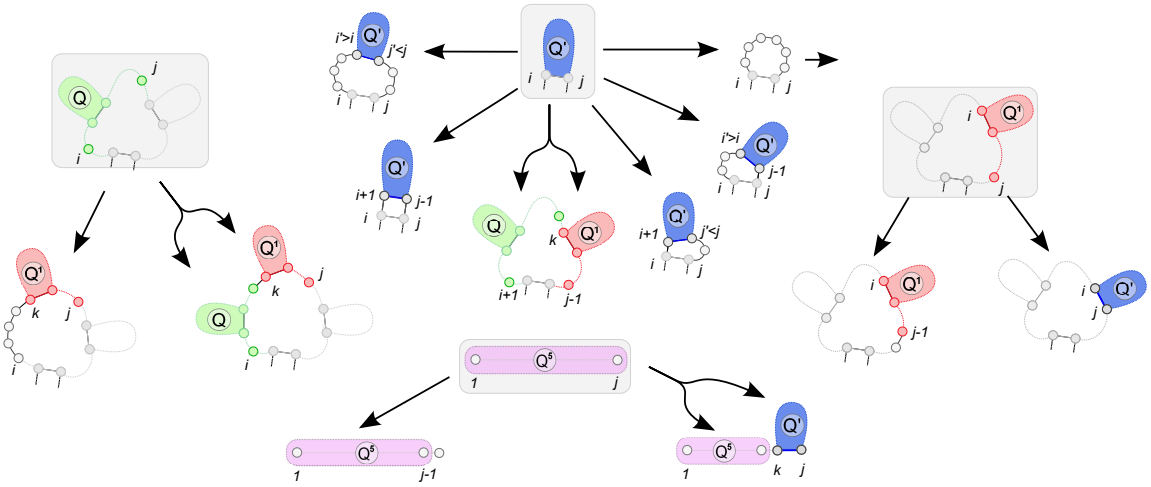

Fig. 2. Simplification of the `Unafold` [32] decomposition of the secondary structures space. Framed states indicate origins of (hyper)arcs.

4.2 RNA secondary structures

Let us first illustrate our approach on RNA secondary structures, for which `Unafold` [32] – the successor of `MFold` [56] – offers an unambiguous scheme. Compared to the original decomposition presented in Markham's thesis [31], the one described in Figure 2 is simplified to ignore dangles.

Proving unambiguity.

- Let us remark that both Q^5 and Q^1 either leave their last base j unpaired (Left), or pairs it to i (Right). Furthermore these two cases are mutually exclusive. Finally Q^1 generates exactly one helix.
- Q always makes at least one call to Q^1 and therefore creates at least one helix. Therefore, it either creates exactly one helix (Left case) or more (Right case), and these two cases are mutually exclusive.
- Q' distinguishes different types of loops. Let m_5, m_3 be the numbers of unpaired bases on the $5'$ strand, $3'$ strand, and h be the number of helices starting from case Q', one can label each of the cases and observe that they are mutually non-overlapping. Namely from left to right, we get the following (m_5, m_3, h) triplets: Interior loop $(>0, >0, 1)$, stacking pair $(0,0,1)$, multiloop $(\geq 0, \geq 0, >1)$, bulges $5'$ $(>0,0,1)$ and $3'$ $(0,>0,1)$, and hairpin loop $(>0,>0,0)$.

Deriving completeness. From previous work by Waterman [54], we know that the generating function of secondary structures with at least one unpaired base between paired bases ($\theta = 1$) is

$$S(z) = \frac{1 - z + z^2 - \sqrt{1 - 2z - z^2 - 2z^3 + z^4}}{2z^2}. \tag{11}$$

Following the general principle of the so-called DSV methodology (See Lorenz *et al* [29] for a presentation in a similar context), the `Unafold` decomposition can be translated into a system of algebraic equations. Namely, one simply replaces any occurrence of k unpaired base with z^k, each basepair with z^2, and any vertex with its associated generating function. Let $Q^5(z)$, $Q(z)$, $Q'(z)$ and $Q^1(z)$ be the generating functions counting the F-paths generated from Q^5, Q, Q' and Q^1 respectively:

$$\begin{aligned}
Q^5(z) &= Q^5(z)\cdot z + Q^5(z)\cdot Q'(z) \qquad Q(z) = \mathrm{Seq}(z)\cdot Q^1(z) + Q(z)\cdot Q^1(z) \qquad Q^1(z) = z\cdot Q^1(z) + Q'(z)\\
Q'(z) &= z^2\cdot \mathrm{Seq}^+(z)\cdot Q'(z)\cdot \mathrm{Seq}^+(z) + z^2\cdot Q'(z) + z^2\cdot Q(z)\cdot Q'(z)\\
&\quad + z^2\cdot Q'(z)\cdot \mathrm{Seq}^+(z) + z^2\cdot \mathrm{Seq}^+(z)\cdot Q^1(z) + \mathrm{Seq}^+(z)\\
\mathrm{Seq}^+(z) &= z\cdot \mathrm{Seq}(z) \qquad \mathrm{Seq}(z) = z\cdot \mathrm{Seq}(z) + 1.
\end{aligned}$$

Solving the system yields $Q^5(z) = S(z)$ which, in conjunction with the unambiguity of the decomposition, proves its completeness.

Application	Algorithm	Weight fun.	Time	Memory	Ref.
A – Energy minimization	Minimal weight	$\pi_{\mathcal{J}}$	$O(n^3)$	$O(n^2)$	[56]
B – Partition function	Weighted count	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(n^3)$	$O(n^2)$	[35]
C – Base-pairing probabilities	Arc-traversal prob.	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(n^3)$	$O(n^2)$	[35]
D – Statistical sampling (k-samples)	Weighted random gen.	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(n^3 + k \cdot n \log n)$	$O(n^2)$	[12, 41]
E – Moments of energy (Mean, Var.)	Moments extraction	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(n^3)$	$O(n^2)$	[36]
F – m-th moment of additive features	Moments extraction	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(m^3 \cdot n^3)$	$O(m \cdot n^2)$	–
G – Correlations of additive features	Moments extraction	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(n^3)$	$O(n^2)$	–

Table 1. Reformulations of secondary structure applications as F-graphs problems and associated complexities.

Applicability of generic algorithms. Let us show that $\mathscr{H}$ fulfills the prerequisites of our algorithms. First it is easily verified that $\mathscr{H}$ is an F-graph. Associating a region $[i, j]$ (resp. $[1, j]$) with each vertex $q^1_{i,j}$, $q_{i,j}$ and $q'_{i,j}$ (resp. q^5_j), one easily verifies that for any F-arc $e \in E$ the width of any region in the head $\mathbf{h}(e)$ is strictly smaller than that of the tail $t(e)$, and the **acyclicity** of $\mathscr{H}$ directly follows. Furthermore, any two vertices in the head $\mathbf{h}(e)$ have non-overlapping associated regions. Consequently $\mathscr{H}$ is **independent**, and a direct application of our generic algorithms gives a set of algorithms summarized in Table 1. This gives a family of efficient $O(n^3)$ algorithms for assessing RNA secondary structure properties at the Boltzmann equilibrium.

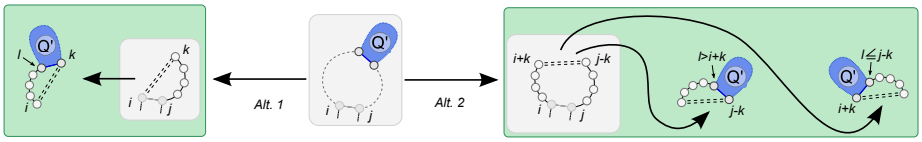

Fig. 3. Alternative exhaustive strategies for interior loops.

Remark 4: In interior loops, the set of F-arcs generated for the Q' case has apparent cardinality in $O(n^4)$. This can be brought back to $O(n^3)$ by enforcing constraints on the energy function. Traditionally, the accepted practice is to bound the interior loop $size\,(j'-j)+(i'-i)$ from above by a predefined constant $K \approx 30$. Exhaustive $O(n^3)$ decompositions can also be proposed (Figure 3) by decomposing the internal loop into additively-contributing regions. A first option may generate independently the left and right unpaired regions (Figure 3, Left), while an alternative may decompose internal loops into a symmetric loop followed by a fully asymmetric one (Figure 3, Right).

4.3 Simple-type pseudoknots

In his seminal work, Akutsu [1] focused on a subset of pseudoknots motifs, the simple-type pseudoknots, and proposed algorithms of complexity in $O(n^4)$ for simple non-recursive pseudoknots in a basepair-maximisation energy model, and in $O(n^5)$ for recursive pseudoknots and loop-based energy models. However, the decomposition proposed in [1] is **ambiguous**, e.g. there exists different ways to create unpaired regions. Therefore we propose in Figure 4 an unambiguous decomposition for the same conformation space.

Previous results. In a previous work [47, 48], one of the authors showed that simple-type pseudoknots can be encoded by a simple formal language, in bijection with a context-free language. Here we focus on partly recursive simple pseudoknots presented in Figure 4. They can be encoded by a well-parenthesized word p over two systems of parentheses $\{(f, \bar{f}), (g, \bar{g})\}$, respectively indicating the leftmost and rightmost basepairs in Figure 4, and an unpaired character c such that

$$p = (c^* f)^n\, p'\, (g\, c^*)^{m_1}\, (\bar{f}\, c^*)^{n_1}\, (g\, c^*)^{m_2}\, (\bar{f}\, c^*)^{n_2} \cdots (g\, c^*)^{m_k}\, (\bar{f}\, c^*)^{n_k - 1}\, \bar{f}\, p''\, \bar{g}\, (c^*\, \bar{g})^{m-1} \tag{12}$$

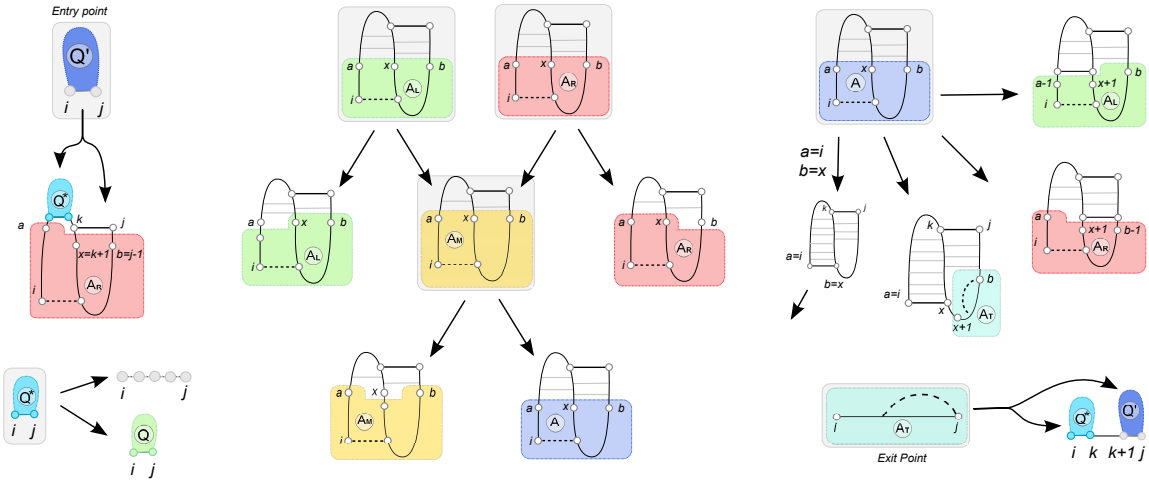

Fig. 4. An unambiguous decomposition for simple non-recursive pseudoknots that captures the Akutsu/Uemura class of pseudoknots. This decomposition yields $O(n^4)/\Theta(n^4)$ time/memory algorithms for partially recursive pseudoknots and can be extended to include recursive pseudoknots and/or Turner energy contributions in $O(n^5)/\Theta(n^4)$.

where k is some integral value, $\sum_{i=1}^{k} n_i = n \geq 1$, $\sum_{i=1}^{k} m_i = m \geq 1$, and p', p'' are any two recursively-generated conformations.

Completeness. Let us show that the decomposition in Figure 4 is complete, i.e. that any partially recursive pseudoknot can be generated by the decomposition.

Let us initially focus on base-pairs and ignore unpaired bases. The smallest word within the language of Equation 12 is $f\,p'\,g\,\bar{f}\,p''\,\bar{g}$ which can be generated by applying the initial case ($Q \to A_L \to A_M \to A \rightsquigarrow p' \ldots g \ldots \bar{g}$) followed directly by the terminal case ($A \to A_T \rightsquigarrow f\,p'\,g\,\bar{f}\,p''\,\bar{g}$). Moreover through a sequence $A \to A_R \to A_M \to A$, one adds an outermost edge around the right part $g \ldots \bar{g}$. So through m iterations of the sequence the decomposition generates any structure $g^{m_1} \ldots \bar{g}^{m_1}$. Similarly through a sequence $A \to A_L \to A_M \to A$ one adds an outermost edge around the left part $f \ldots \bar{f}$, and after n_1 iterations any structure $f^{n_1} \ldots \bar{f}^{n_1}$ is generated. Since these two sequences can be combined and alternated (starting with the initial case and finishing with the terminal case), then the decomposition generates any word

$$p = f^n\,p'\,g^{m_1}\,\bar{f}^{n_1}\,g^{m_2}\,\bar{f}^{n_2} \cdots g^{m_k}\,\bar{f}^{n_k}\,p''\,\bar{g}^m\,\bar{g}. \tag{13}$$

For the recursive call p', it is easily verified that Q^* generates any (PK) structure. For p'' it is worth mentioning that, at a base-pairing level, $A \to A_T$ (right base paired) and $A \to \emptyset$ cover all possible situations.

Arbitrary numbers of unpaired bases c can also be inserted right before the opening f of a leftward base pair (resp. after closure $\bar{f}$ of a leftward base pair, after the opening g of a right base pair and before the closure $\bar{g}$ of a right base pair) by repeatedly applying the $A_L \to A_L$ (resp. $A_M \to A_M$, $A_L \to A_L$ and $A_M \to A_M$) rule after adding a left (resp. right) base pair. Consequently any structure described by a word in Equation 12 can be generated by the decomposition.

Unambiguity. Let us now address the unambiguity of the decomposition, using our approach based on generating functions. Equation 12 immediately gives a system of equations relating $AU(z)$, the generating function of simple partially recursive pseudoknots, to $S(z)$ the gen. fun. of all structures:

$$AU(z) = \sum_{k\geq 1} \left(\frac{z}{1-z}\right)^n S(z) \left(\frac{z}{1-z}\right)^{m_1} \left(\frac{z}{1-z}\right)^{n_1} \cdots \left(\frac{z}{1-z}\right)^{n_k-1} z\,S(z)\,z\left(\frac{z}{1-z}\right)^{m-1} = \frac{z^4\,S(z)^2\,(1-z)}{1-2\,z-z^2}.$$

Now consider the dynamic programming decomposition illustrated by Figure 4. Associating generating functions to each type of vertices and translating assigned bases into monomials, we obtain the following system of equations:

$$Q'(z) = z^2\,S(z)\,A_R(z) \qquad A_L(z) = z\,A_L(z) + A_M(z) \qquad A_R(z) = z\,A_R(z) + A_M(z)$$
$$A_M(z) = z\,A_M(z) + A(z) \qquad A(z) = z^2\,A_R(z) + z^2\,A_L(z) + z^2\,S(z) \qquad A_T(z) = S(z)(1-z) - 1.$$

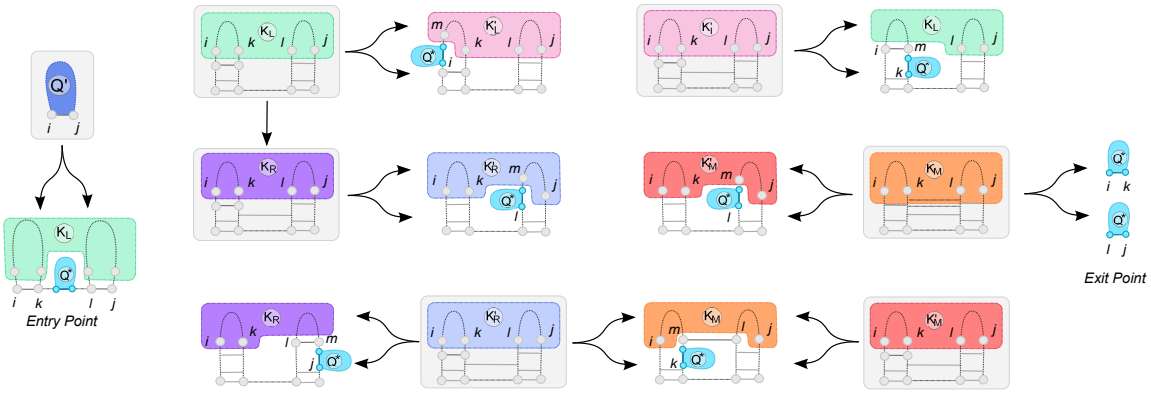

Fig. 5. Unambiguous decomposition of fully recursive kissing hairpins.

The last expression for $A_T(z)$ follows directly from the observation that any structure in Q can be written as a sequence of structures from Q' interleaved with sequences of unpaired bases. Given that A_T cannot feature unpaired bases on its right end, one of the sequence of unpaired base must be removed. Furthermore A_T does not generate the empty structure, so we have $S(z) = (A_T(z)+1)/(1-z)$. Solving the system gives $Q'(z) = \frac{S^2(z)z^4(1-z)}{1-2z+z^2} = AU(z)$ and the unambiguity/correctness of the decomposition directly follow.

4.4 Fully-recursive kissing hairpins

Kissing hairpins (KH) are pseudoknotted structure composed of two helices whose terminal loops are linked by a third helix. These pseudoknots are frequently observed, and are exhaustively predicted by Chen *et al* [8] in time complexity in $O(n^5)$, and in $O(n^3)/O(n^4)$ under restrictions by Theis *et al* [50]. Figure 5 presents an unambiguous decomposition which generates the space of recursive kissing hairpins.
Previous results. Again, an encoding of kissing hairpins can be found in earlier work by one of the authors [47], showing that any KH pseudoknot can be represented by a word p over three systems of parentheses $\{(f,\bar{f}),(g,\bar{g}),(h,\bar{h})\}$ (respectively denoting leftmost, central and rightmost helices) such that:

$$p = (fS)^n (gS)^m (\bar{f}S)^n (hS)^k (\bar{g}S)^m (\bar{h}S)^{k-1} \bar{h}. \tag{14}$$

Completeness. First let us remark that the *minimal* conformation generated by the decomposition is $K_L \to K_R \to K'_R \to K_M \rightsquigarrow fSgS\bar{f}ShS\bar{g}S\bar{h}$. Remark that one can iterate arbitrarily over the states $K_L \to K'_L \to K_L$, $K'_R \to K_R \to K'_R$ and $K'_M \to K_M \to K_M$. Consequently one may *insert* patterns $(K_L \to K'_L \to K_L)^{n-1} \rightsquigarrow (Sf)^{n-1}\cdots(\bar{f}S)^{n-1}$, $(K'_R \to K_R \to K'_R)^{k-1} \rightsquigarrow (hS)^{k-1}\cdots(\bar{h}S)^{k-1}$ and $(K_M \to K'_M \to K_M)^m \rightsquigarrow (gS)^{m-1}\cdots(S\bar{g})^{m-1}$ in the minimal word above, and produce any conformation denoted by

$$f(Sf)^{n-1}S(gS)^{m-1}yS(\bar{f}S)^{n-1}\bar{f}ShS(hS)^{k-1}\bar{g}(S\bar{g})^{m-1}S(\bar{h}S)^{k-1}\bar{h}$$

where one recognizes the language of Equation 14 upon simple expansion.
Unambiguity. Equation 14 allows to derive the generating function $KH(z)$ of kissing-hairpin as a function of $S(z)$ the gen. fun. of all structures:

$$KH(z) = \sum_{n,m,k\geq 1} (zS(z))^n (zS(z))^m (zS(z))^n (zS(z))^k (zS(z))^m (zS(z))^{k-1} z = \frac{z^6 S(z)^5}{(1-z^2S(z)^2)^3}. \tag{15}$$

Now consider the dynamic programming decomposition illustrated by Figure 5, and translate it into a system of functional equation:

$$K(z) = z^4 K_L(z) S(z)$$

$$K_L(z) = S(z)K'_L(z) + K_R(z) \qquad K'_L(z) = z^2 K_L(z)S(z) \qquad K_M(z) = K'_M(z)S(z) + S(z)^2$$

$$K'_M(z) = z^2 K_M(z)S(z) \qquad K_R(z) = K'_R(z)S(z) \qquad K'_R(z) = z^2 K_R(z)S(z) + z^2 K_M(z)S(z)$$

Application	Algorithm	Weight fun.	Time	Memory	Ref.
Simple type pseudoknots (Akutsu&Uemura)					
A – Energy minimization	Minimal weight	π_{bp}	$O(n^4)$	$O(n^4)$	[1]
B – Partition function	Weighted count	$e^{\frac{-\pi_{bp}}{RT}}$	$O(n^4)$	$O(n^4)$	[6, 7] in $\Theta(n^6)$
C – Base-pairing probabilities	Arc-traversal prob.	$e^{\frac{-\pi_{bp}}{RT}}$	$O(n^4)$	$O(n^4)$	–
D – Statistical sampling (k-samples)	Weighted rand. gen.	$e^{\frac{-\pi_{bp}}{RT}}$	$O(n^4 + k \cdot n \log n)$	$O(n^4)$	–
E – Moments of energy (Mean, Var.)	Moments extraction	$e^{\frac{-\pi_{bp}}{RT}}$	$O(n^4)$	$O(n^4)$	–
F – m-th moment of additive features	Moments extraction	$e^{\frac{-\pi_{bp}}{RT}}$	$O(m^3 \cdot n^4)$	$O(m \cdot n^4)$	–
Fully recursive Kissing Hairpins					
A – Energy minimization	Minimal weight	$\pi_{\mathscr{T}}$	$O(n^5)$	$O(n^4)$	[8]
B – Partition function	Weighted count	$e^{\frac{-\pi_{\mathscr{T}}}{RT}}$	$O(n^5)$	$O(n^4)$	–
C – Base-pairing probabilities	Arc-traversal prob.	$e^{\frac{-\pi_{\mathscr{T}}}{RT}}$	$O(n^5)$	$O(n^4)$	–
D – Statistical sampling (k-samples)	Weighted rand. gen.	$e^{\frac{-\pi_{\mathscr{T}}}{RT}}$	$O(n^5 + k \cdot n \log n)$	$O(n^4)$	–
E – Moments of energy (Mean, Var.)	Moments extraction	$e^{\frac{-\pi_{\mathscr{T}}}{RT}}$	$O(n^5)$	$O(n^4)$	–
F – m-th moment of additive features	Moments extraction	$e^{\frac{-\pi_{\mathscr{T}}}{RT}}$	$O(m^3 \cdot n^5)$	$O(m \cdot n^4)$	–

Table 2. Summary of ensemble based algorithms on simple pseudoknots and kissing hairpins. π_{bp} stands for the simple Nussinov-Jacobson energy model, and $\pi_{\mathscr{T}}$ for a Turner-like model based on loops contributions.

Solving the system gives $K(z) = \frac{z^6 S(z)^5}{(1-z^2 S(z)^2)^3} = KH(z)$ and the unambiguity of the decomposition immediately follows. Again hypergraphs algorithms can be used, and specialize into the complexities summarized in Table 2.

5 Extending the framework: Extraction of moments and exact correlations

A last application addresses the extraction of statistical measures for **additive features**. Let us first define a **feature** as a function $\alpha : E \to \mathbb{R}^+$ extended additively over F-paths such that $\alpha(p) = \sum_{e \in p} \alpha(e)$. One may then want to characterize the distribution of a random variable $X = \alpha(p)$, for $p \in \mathscr{P}$ a random F-path drawn according to the weighted distribution. As it is not necessarily feasible to determine the exact distribution of X, one can examine statistical measures such as its

$$\text{Mean } \mu_X = \mathbb{E}[X] \quad \text{and} \quad \text{Variance } \text{Var}_X = \mathbb{E}[X^2] - \mu_X^2,$$

e.g. from which the distribution is fully determined in the case of Gaussian distributions. Even when the distribution is not normal, it can still be characterized by a list of measures called **moments** of X, the m-th moment being defined as $\mathbb{E}[X^m] = \sum_{p \in \mathscr{P}} \alpha(p)^m \cdot \pi(p)/w_s$.

Moreover in the presence of multiple features $(X_1 := \alpha_1(p), \ldots, X_k := \alpha_k(p))$, similar measures can be used to estimate their level of dependency. One such measure is the **Pearson product-moment correlation coefficient** ρ_{X_1,X_2} defined for two random variables as

$$\rho_{X_1,X_2} = \frac{\text{Cov}_{X_1,X_2}}{\sqrt{\text{Var}_{X_1} \cdot \text{Var}_{X_2}}} = \frac{\mathbb{E}[X_1 \cdot X_2] - \mathbb{E}[X_1] \cdot \mathbb{E}[X_2]}{\sqrt{\text{Var}_{X_1} \cdot \text{Var}_{X_2}}}$$

The correlation above involves the expectation of a product of two random variables which is an instance of a **generalized moment**, defined for the set of F-paths starting from $s \in V$ as

$$\mathbb{E}[X_1^{m_1} \cdots X_k^{m_k} \mid s] = \sum_{p \in \mathscr{P}_s} \frac{\pi(p)}{w_s} \prod_{i=1}^{k} \alpha_i(p)^{m_i}. \tag{16}$$

Extracting such moments can be quite useful, allowing one to get access to average properties of structures (#Hairpins, #Occurrences of pseudoknots…) and their correlations within a weighted ensemble. For instance, Miklos *et al* [36] proposed an $\mathscr{O}(m^2 \cdot n^3)$ algorithm for computing the m-th moment of

the Energy distribution for secondary structure in order to compare the distribution of free-energy in non-coding RNAs and random sequences. We are going to show how these generalized moments can be extracted directly through a generalization of the weighted count algorithm.

Theorem 1. *Let $\alpha := (\alpha_1, \cdots, \alpha_k)$ be a vector of additive features and $\mathbf{m} := (m_1, \cdots, m_k)$ be a k-tuple of natural integers. Then the pseudo-moment $c_s^{\mathbf{m}} := \mathbb{E}[X_1^{m_1} \cdots X_k^{m_k} \mid s] \cdot w_s$ of α in a weighted distribution can be recursively computed through*

$$c_s^{\mathbf{m}} = \sum_{e=(s\to\mathbf{t})} \pi(e) \cdot \sum_{\substack{\mathbf{m}', \left(\mathbf{m}''_1, \cdots, \mathbf{m}''_{|t|}\right) \\ s.\,t.\ \mathbf{m}' + \sum_j \mathbf{m}''_j = \mathbf{m}}} \prod_{i=1}^{k} \binom{m_i}{m'_i, m''_{1,i}, \cdots, m''_{|t|,i}} \cdot \alpha_i(e)^{m'_i} \cdot \prod_{i=1}^{|t|} c_{t_i}^{\mathbf{m}''_i} \tag{17}$$

in $\mathcal{O}\left((|E|+|V|) \cdot k \cdot t^+ \cdot \prod_{i=1}^{k} m_i^{t^+ + 1}\right)$ time complexity and $\Theta\left(|V| \cdot \prod_{i=1}^{k} m_i\right)$ memory where $t^+ = \max_{(s\to t)\in E}(|t|)$ is the maximal out-degree of an arc.

Adding this new generic algorithms automatically creates new applications for each an every conformation space as summarized in Figure 2. This simultaneous extension – for all conformational spaces – of possible ensemble applications constitues in our opinion one of the main benefit of detaching the decomposition from its exploration.

6 Conclusion and Perspectives

In this paper, we established the foundation of a combinatorial approach to the design of algorithms for complex conformation spaces. We built on an hypergraph model introduced in the context of RNA secondary structure by Finkelstein and Roytberg [16], which we extended in several direction. First we formulated classic and novel generic algorithms on Forward-Hypergraphs for weighted ensembles, allowing one to derive base-pairing probabilities, perform statistical sampling and extract moments of the distribution of additive features. Then we showed how combinatorial arguments based on generating functions could be used to simplify the proof of correctness for designed decompositions. We illustrated the full programme on classic secondary structures, simple type pseudoknots and fully-recursive kissing hairpin pseudoknots for which we provided decompositions that were proven to be unambiguous and complete with respect to previous work. The hypergraph formulation of the decomposition, coupled with the generic algorithms, readily gave a family of novel algorithms for complex – yet relevant – conformation spaces.

Let us mention some perspectives to our contribution. Firstly the principles and algorithms described here could easily be implemented as a general *compiler* tools for F-Graphs algorithms. Such a compiler could be coupled with helper tools expanding hypergraphs from succinct descriptions, such as context-free grammars (related to ADP [19]), or M. Möhl's split types [37]. More complex search space could also be modeled, such as those relying on a more detailed representation of RNA structure (e.g. MCFold's NCMs [40]), those capturing RNA-RNA interactions [2, 24], those offering simultaneous alignment and folding (Sankoff's algorithm [46]) or performing mutations on the sequence [53]. Finally our hypergraph framework is not necessarily limited to polynomial algorithms, and algorithmic developments could be proposed to address some of the current algorithmic issues in RNA (inverse folding [3], kinetics [49]) for which no exact polynomial algorithms are currently known (or suspected). More generally it is our hope that, by simplifying and modularizing the process of developing new – algorithmically tractable – conformation spaces, our contribution will help design better, more topologically-realistic[52, 28, 44], energy and conformational spaces to better understand and predict the structure(s) of RNA.

Acknowledgement

The authors wish to express their gratitude to M. Roytberg for pointing out his work on hypergraphs as a unifying framework, and to R. Backofen, M. Möhl and S. Will for fruitful discussions. This research was supported by the Digiteo project "RNAomics". YP was funded by an ANR grant MAGNUM (ANR 2010 BLAN 0204).

References

1. Tatsuya Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Appl. Math.*, 104(1-3):45–62, 2000.
2. C. Alkan, E. Karakoç, J. H. Nadeau, S. C. Sahinalp, and K. Zhang. RNA-RNA Interaction Prediction and Antisense RNA Target Search. In *Proceedings of RECOMB'05*, 2005.
3. M. Andronescu, A. P. Fejes, F. Hutter, H. H. Hoos, and A. Condon. A New Algorithm for RNA Secondary Structure Design. *J Mol Biol*, 336(3):607–624, 2004.
4. M. Bekaert, L. Bidou, A. Denise, G. Duchateau-Nguyen, J. Forest, C. Froidevaux, I. Hatin, J. Rousset, and M. Termier. Towards a computational model for −1 eukaryotic frameshifting sites. *Bioinformatics*, 19:327–335, 2003.
5. M. Bousquet-Mélou and Y. Ponty. Culminating paths. *Discrete Mathematics and Theoretical Computer Science*, 10(2):125–152, 2008.
6. S. Cao and S. J. Chen. Predicting RNA pseudoknot folding thermodynamics. *Nucleic Acids Res*, 34(9):2634–2652, 2006.
7. S. Cao and S-J Chen. Predicting structured and stabilities for H-type pseudoknots with interhelix loop. *RNA*, 15:696–706, 2009.
8. Ho-Lin Chen, Anne Condon, and Hosna Jabbari. An O(n(5)) algorithm for MFE prediction of kissing hairpins and 4-chains in nucleic acids. *Journal of Computational Biology*, 16(6):803–815, 2009.
9. A. Condon, B. Davy, B. Rastegari, S. Zhao, and F. Tarrant. Classifying RNA pseudoknotted structures. *Theoretical Computer Science*, 320(1):35–50, 2004.
10. A. Denise, Y. Ponty, and M. Termier. Controlled non uniform random generation of decomposable structures. *Theoretical Computer Science*, 411(40-42):3527–3552, September 2010.
11. Y. Ding, C. Y. Chan, and C. E. Lawrence. RNA secondary structure prediction by centroids in a boltzmann weighted ensemble. *RNA*, 11:1157–1166, 2005.
12. Y. Ding and E. Lawrence. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Res*, 31(24):7280–7301, 2003.
13. R.M. Dirks and N.A. Pierce. A partition function algorithm for nucleic acid secondary structure including pseudoknots. *J Comput Chem*, 24:1664–1677, 2003.
14. Chuong B Do, Daniel A Woods, and Serafim Batzoglou. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, 22(14):e90–e98, Jul 2006.
15. F. Ferrè, Y. Ponty, W. A. Lorenz, and Peter Clote. DIAL: A web server for the pairwise alignment of two RNA 3-dimensional structures using nucleotide, dihedral angle and base pairing similarities. *Nucleic Acids Res*, (35 (Web server issue)):W659–668, July 2007.
16. A. V. Finkelstein and M. A. Roytberg. Computation of biopolymers: a general approach to different problems. *Biosystems*, 30(1-3):1–19, 1993.
17. P. Flajolet, P. Zimmermann, and B. Van Cutsem. Calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132:1–35, 1994. A preliminary version is available in INRIA Research Report RR-1830.
18. Philippe Flajolet. Analytic models and ambiguity of context-free languages. *Theoretical Computer Science*, 49:283–309, 1987.
19. R. Giegerich. A systematic approach to dynamic programming in bioinformatics. *Bioinformatics*, 16(8):665–677, Aug 2000.
20. Michiaki Hamada, Hisanori Kiryu, Kengo Sato, Toutai Mituyama, and Kiyoshi Asai. Prediction of RNA secondary structure using generalized centroid estimators. *Bioinformatics*, 25(4):465–473, Feb 2009.
21. Arif Ozgun Harmanci, Gaurav Sharma, and David H Mathews. Stochastic sampling of the rna structural alignment space. *Nucleic Acids Res*, 37(12):4063–4075, Jul 2009.
22. Ivo L Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Res*, 31(13):3429–3431, Jul 2003.
23. Fenix W D Huang, Wade W J Peng, and Christian M Reidys. Folding 3-noncrossing rna pseudoknot structures. *J Comput Biol*, 16(11):1549–1575, Nov 2009.
24. Fenix W D Huang, Jing Qin, Christian M Reidys, and Peter F Stadler. Target prediction and a statistical sampling algorithm for RNA-RNA interaction. *Bioinformatics*, 26(2):175–181, Jan 2010.
25. G. Kucherov, L. Noe, and Y. Ponty. Estimating seed sensibility on homogenous alignments. In IEEE, editor, *Proceedings of Fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE'04)*, page 387, 2004.
26. F. Lefebvre. A grammar-based unification of several alignment and folding algorithms. In *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, pages 143–154. AAAI Press, 1996.
27. F. Lefebvre. *Grammaires S-attribuées multi-bandes et applications à l'analyse automatique de séquences biologiques*. PhD thesis, École Polytechnique, 1997.
28. A. Lescoute and E. Westhof. Topology of three-way junctions in folded RNAs. *RNA*, 12(1):83–93, 2006.
29. W.A. Lorenz, Y. Ponty, and P. Clote. Asymptotics of RNA shapes. *Journal of Computational Biology*, 15(1):31–63, Jan–Feb 2008.

30. R. B. Lyngsø and C. N. S. Pedersen. RNA pseudoknot prediction in energy-based models. *Journal of Computational Biology*, 7(3-4):409–427, 2000.
31. Nicholas R Markham. *Algorithms and software for nucleic acid sequences*. PhD thesis, Faculty of Rensselaer Polytechnic Institute, 2006.
32. Nicholas R Markham and Michael Zuker. UNAFold: software for nucleic acid folding and hybridization. *Methods Mol Biol*, 453:3–31, 2008.
33. D. H. Mathews. Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA*, 10(8):1178–1190, 2004.
34. D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J Mol Biol*, 288:911–940, 1999.
35. J.S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–1119, 1990.
36. István Miklós, Irmtraud M Meyer, and Borbála Nagy. Moments of the boltzmann distribution for RNA secondary structures. *Bull Math Biol*, 67(5):1031–1047, Sep 2005.
37. Mathias Möhl, Sebastian Will, and Rolf Backofen. Lifting prediction to alignment of rna pseudoknots. *J Comput Biol*, 17(3):429–442, Mar 2010.
38. U. Mückstein, I. L. Hofacker, and P. F. Stadler. Stochastic pairwise alignments. *Bioinformatics*, 18 Suppl 2:S153–S160, 2002.
39. R. Nussinov and A. B. Jacobson. Fast algorithm for predicting the secondary structure of single stranded RNA. *Proc. Natl. Acad. Sci. U. S. A.*, 77(11):6309–6313, 1980.
40. M. Parisien and F. Major. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*, 452(7183):51–55, 2008.
41. Y. Ponty. Efficient sampling of RNA secondary structures from the boltzmann ensemble of low-energy: The boustrophedon method. *J Math Biol*, 56(1-2):107–127, Jan 2008.
42. J. Reeder and R. Giegerich. Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics*, 5:104, 2004.
43. Janina Reeder, Peter Steffen, and Robert Giegerich. Effective ambiguity checking in biosequence analysis. *BMC Bioinformatics*, 6:153, 2005.
44. Christian M Reidys, Fenix W D Huang, Jørgen E Andersen, Robert C Penner, Peter F Stadler, and Markus E Nebel. Topology and prediction of rna pseudoknots. *Bioinformatics*, 27(8):1076–1085, Apr 2011.
45. E. Rivas and S.R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J Mol Biol*, 285:2053–2068, 1999.
46. D. Sankoff. Simultaneous solution of the rna folding, alignment and protosequence problems. *SIAM J Appl Math*, 45:810–825, 1985.
47. C. Saule. *Modèles combinatoires des structures d'ARN avec ou sans pseudonœuds, application à la comparaison de structures.* PhD thesis, Université Paris Sud, Ecole doctorale informatique., December 2010.
48. C. Saule, M. Régnier, J-M. Steyaert, and A. Denise. Counting RNA pseudoknotted structures. *Journal of Computational Biology*, To appear.
49. Chris Thachuk, Ján Manuch, Arash Rafiey, Leigh-Anne Mathieson, Ladislav Stacho, and Anne Condon. An algorithm for the energy barrier problem without pseudoknots and temporary arcs. *Pac Symp Biocomput*, pages 108–119, 2010.
50. Corinna Theis, Stefan Janssen, and Robert Giegerich. Prediction of rna secondary structure including kissing hairpin motifs. In *Proceedings of WABI 2010*, pages 52–64, 2010.
51. I. Tinoco, P. N. Borer, B. Dengler, M. D. Levin, O. C. Uhlenbeck, D. M. Crothers, and J. Bralla. Improved estimation of secondary structure in ribonucleic acids. *Nat New Biol*, 246(150):40–41, Nov 1973.
52. G. Vernizzi, P. Ribeca, H. Orland, and A. Zee. Topology of pseudoknotted homopolymers. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 73(3):031902, 2006.
53. Jérôme Waldispühl, Srinivas Devadas, Bonnie Berger, and Peter Clote. Efficient algorithms for probing the RNA mutation landscape. *PLoS Comput Biol*, 4(8):e1000124, 2008.
54. M. S. Waterman. Secondary structure of single stranded nucleic acids. *Advances in Mathematics Supplementary Studies*, 1(1):167–212, 1978.
55. H. S. Wilf. A unified setting for sequencing, ranking, and selection algorithms for combinatorial objects. *Advances in Mathematics*, 24:281–291, 1977.
56. M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res*, 9:133–148, 1981.