A combinatorial framework for the design of (pseudoknotted) RNA algorithms

Yann Ponty* and Cédric Saule

[1] LIX, École Polytechnique/CNRS/INRIA AMIB
yann.ponty@lix.polytechnique.fr
[2] LRI, Université Paris-Sud/XI/INRIA AMIB
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 suspected for novel RNAs. 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. The restriction of potential contacts allowed Nussinov and Jacobson (36) to design a $\Theta(n^3)$ dynamic-programming algorithm to maximize the number of base-pairs. Building on a nearest neighbor model for the free-energy proposed by Tinoco *et al* (48) and extended by the Turner group, Zuker and Stiegler (53) 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 (31). An independent implementation of the algorithm is proposed within the VIENNA package maintained by Hofacker (21). Probabilistic alternatives (SFOLD (10), CONTRAFOLD (13) and CENTROIDFOLD (19)) have also recently been proposed with substantial success, relying on a dynamic programming scheme similar to that of MFOLD to traverse the

* To whom correspondence should be addressed

conformation space in polynomial time coupled with some postprocessing step to elect one or several candidate secondary structure(s).

Ensemble approaches. Since the seminal work of McCaskill (32), the concept of Boltzmann equilibrium studied by statistical mechanics have been used to embrace the diversity of the structural ensemble of an RNA sequence. Namely McCaskill 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 very dynamic-programming scheme used for the MFE folding. Coupled with a variant of the inside/outside algorithm, this led to an exact computation of probabilities for the base-pairs in the Boltzmann-weighted ensemble. Intuitively, this opened the door for more robust predictions, as the MFE folding paradigm may be challenged by RNAs whose MFE is an outlier, or some artefact due to an intrinsically imperfect energy model. This intuition was later validated by Mathews (30) who showed that the Boltzmann probability correlated well with the actual presence of base-pairs in experimentally-determined structures. Ding *et al* (10) pushed this paradigm shift a step further by clustering sets of structures sampled with respect to a Boltzmann distribution, improving on the positive-predictive-value (PPV) of existing algorithms. This ensemble view has naturally spread toward other applications of dynamic-programming in Bioinformatics (sequence alignement (33), simultaneous alignment and folding (20), 3D structural alignement (14)), and is increasingly becoming a part of the standard *algorithmic toolbox* of bioinformaticians.

Pseudoknotted conformations. Although substantially successful in their task, the above-mentioned secondary structure prediction algorithms are intrinsically limited in their predictions by their inability to explore conformations that feature 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 the simple nearest-neighbor model (1; 27). However such conformations do naturally occur, and can be essential to functional mechanisms such as -1-frameshift recoding events (4) or tertiary motifs (37). For these reasons, many exact DP approaches (42; 27; 12; 39; 5; 6; 7; 6; 22; 47; 41) 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 (12; 5; 41)) are based on ambiguous dynamic programming schemes, leading them to consider certain structures multiple times. While such a property does not constitute an issue in the context of energy minimization, this 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 towards certain structures.

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* (8) that many PK conformational spaces can be modeled as a formal language, and Flajolet (17) has shown, using a combinatorial argument, that even simple context-free languages can be inherently ambiguous, i.e. may not be generated by any unambiguous context-free grammar. A second tentative explanation is more historical: DP algorithms designers were initially focused on optimisation problems, and considered the DP equation, not the decomposition of the search space, to be the central object of their contributions. Indeed in the optimisation 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 higher-level energy terms and topological limitations, these 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 beyond context-free languages.

Our proposal: 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 leave to an automated procedure the actual production of codes/algorithms for specific applications. In that, we share the philosophy underlying Lefebvre's multi-tape attributed grammars (24) and Giegerich's Algebraic Dynamic Programming (ADP) (18). However it is crucial for our formalism to capture contextual aspects of pseudoknots, an aspect that was not central to the development of the two above frameworks and therefore partially addressed. For these reasons, we chose to build

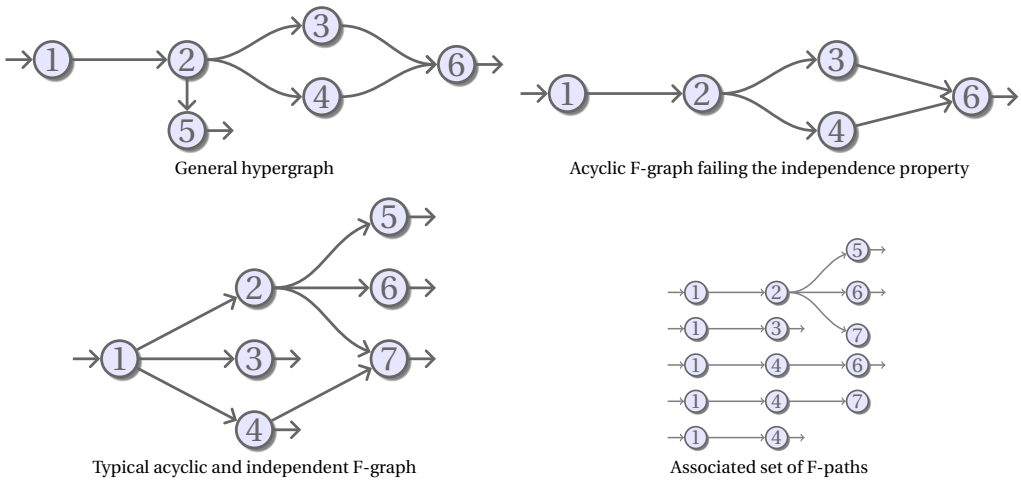

Fig. 1. Illustration of F-Graphs, F-Paths and Independence property. Straight lines indicate classic arcs, and bent curves indicate hyperarcs.

on and revisit an hypergraph analogy proposed by Finkelstein *et al* (15) as a unifying framework for RNA folding and other applications of Dynamic Programming in bioinformatics.

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 (29), (weighted) base-pair maximisation version of Akutsu's simple-type pseudoknots (1) and fully-recursive kissing hairpins (Unambiguous restriction of Chen *et al* (7)). 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* (35)). 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 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 $n \in V'$, $(v_n \rightarrow \mathbf{c}_n) \in E$. For the sake of simplicity, we may omit the implicit V' and denote an F-path by 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 $(s \rightarrow \varnothing) \in E$, or $\langle s, D_1 \dots D_{|\mathbf{t}|} \rangle$ if $(s \rightarrow \mathbf{t}) \in E$ 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 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 $\pi : E \rightarrow \mathbb{R}^+$ 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 $l \rightarrow \varnothing$ 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 (52) 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 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 (9) 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 w_s of F-paths starting from a given vertex s.

$$w_s = \sum_{p'\in\mathscr{P}_{v_0}} \pi(p') = \sum_{e=(s\to\mathbf{h}(e))\in E} \pi(e) \cdot \prod_{s'\in\mathbf{h}(e)} w_{s'}, \quad \forall s \in V \tag{5}$$

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 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 generation of a path p can be achieved 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 (16; 38) 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: 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 $t(e)$; A **support edge** $e = (t(e) \to \mathbf{h}(e))$; 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

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

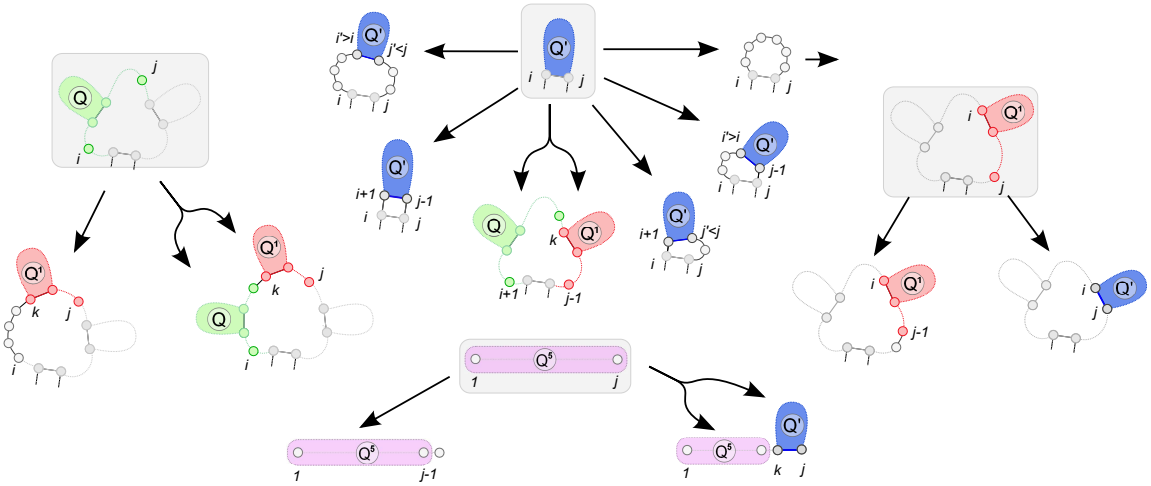

Fig. 2. Simplification of the `Unafold` (29) decomposition of the secondary structures space. Framed states indicate origins of (hyper)arcs.

4 F-graphs reformulation of (Pseudoknotted) RNA folding search spaces

4.1 Foreword: Shortening correctness proofs through generating functions

Unambiguity is a necessary property for ensemble applications of dynamic programming, requiring each element of the search space to be traversed (or equivalently generated) at most once. Since this notion is intimately related to the semantics associated to the F-paths, it cannot be tackled in an automated way at the decomposition level[3] but will usually require user-assigned semantics, from which local disjointness can be asserted. On the other hand, the **completeness** of a decomposition requires to cover, or conversely generate, the entire search space. This property ensures that each element of the search space is traversed at most once. Again, completeness is usually proven in a non-automated way by proving that any element of the search space can be *parsed* by the proposed decomposition.

Proving the **correctness** of ensemble DP schemes will typically require proving both completeness and unambiguity. Such a task may become challenging for complex decompositions. In order to simplify the validation and therefore the design of new search spaces, we propose a proof technique based on generating functions. Indeed proving the correctness is equivalent to establishing that the structures $\mathscr{S}_{\mathscr{D}}$ generated by the decomposition are in bijection with the targeted search space.

However when information is available on cardinality for both the search space generated by the decomposition and the targeted one, it is no longer necessary to prove both conditions, since one will naturally imply the other. Therefore we propose an alternative approach based on generating functions. Namely $\mathscr{S}$ be the targeted search space, $\mathscr{D}$ be a decomposition, $\mathscr{S}_{\mathscr{D}} \subset \mathscr{S}$ the (multi)set of objects generated from $\mathscr{D}$, and $|\cdot| : \mathscr{S} \to \mathbb{N}$ be a any (size) function. Let $S(z) = \sum_{o \in \mathscr{S}} z^{|o|}$ and $S_{\mathscr{D}}(z) = \sum_{o' \in \mathscr{S}_{\mathscr{D}}} z^{|o'|}$ be the generating functions of $\mathscr{S}$ and $\mathscr{S}_{\mathscr{D}}$ respectively, then the following proposition hold.

Proposition 1 (Completeness). *Assume that $\mathscr{D}$ is unambiguous, then $\mathscr{D}$ is complete iff $S(z) = S_{\mathscr{D}}(z)$.*

Proposition 2 (Unambiguity). *Assume that $\mathscr{D}$ is complete, then $\mathscr{D}$ is unambiguous iff $S(z) = S_{\mathscr{D}}(z)$.*

4.2 RNA secondary structures

Let us first illustrate our approach on RNA secondary structures, for which `Unafold` (29) – the successor of `MFold` (53) – offers an unambiguous scheme. Compared to the original decomposition presented in Markham's thesis (28), the one described in Figure 2 is simplified to ignore dangles.

[3] Algebraic Dynamic Programming partially addresses this issue, and the interested reader is referred to an early contribution by Reeder *et al* (40).

Proving unambiguity. Let us prove the unambiguity of `Unafold` as outlined in Markham's thesis (28):

- 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 (51), 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* (26) for a presentation in a similar context), the `Unafold` decomposition can be translated into a grammar. The unambiguity of the decomposition carries to the grammar, which can then be transformed into a system of algebraic equations involving $Q^5(z)$, $Q(z)$, $Q'(z)$ and $Q^1(z)$ the generating functions counting the secondary structures generated from $Q^5(z)$, Q, Q' and Q^1 respectively:

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

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

F-graph expansion. Within a sequence of size n, we obtain an F-graph $\mathscr{H} = (v_0, V, E, \pi)$ where: The set of vertices V is simply the cross product of the sequence intervals and cases,

$$V := \left\{q_{i,j}, q^1_{i,j}, q'_{i,j}\right\}_{0\leq i<j\leq n} \bigcup \left\{q^5_j\right\}_{0\leq j\leq n};$$

The initial state is $v_0 = q^5_n$; and the set of F-arcs E is derived from Figure 2 by translating the productions into F-arcs for each coherent combination of indices:

- Q: $\left\{q_{i,j} \to q^1_{k,j}\right\}_{1\leq i\leq k<j\leq n}$ and $\left\{q_{i,j} \to q_{i,k-1}\, q^1_{k,j}\right\}_{1\leq i\leq k<j\leq n}$
- Q^1: $\left\{q^1_{i,j} \to q^1_{i,j-1}\right\}_{1\leq i<j\leq n}$ and $\left\{q^1_{i,j} \to q'_{i,j}\right\}_{1\leq i<j\leq n}$
- Q^5: $\left\{q^5_j \to q^5_{j-1}\right\}_{1<j\leq n}$, $\left\{q^5_j \to q^5_{k-1} q'_{k,j}\right\}_{1\leq k<j\leq n}$ and $\{q^5_1 \to \varnothing\}$.
- Q': $\left\{q'_{i,j} \to q'_{i',j'}\right\}_{1\leq i<i'<j'<j\leq n}$, $\left\{q_{i,j} \to q_{i,k-1}\, q^1_{k,j}\right\}_{1\leq i\leq k<j\leq n}$ and $\left\{q'_{i,j} \to \varnothing\right\}$ provided that $(j-i-1) \geq \theta$ and (i,j) can form a basepair. No F-arc otherwise.

Since the F-arcs exactly correspond to the right-hand side alternatives in the decomposition, the ability of our model to capture the full Turner model (dangles put asides) derives from that of `Unafold`. Namely we associate with each F-arc, through a **Turner weight function** $\pi_{\mathscr{T}} : E \to \mathbb{R}$, the free-energy contribution of the associated loop in the Turner model.

Application	Algorithm	Weight fun.	Time	Memory	Ref.
A – Energy minimization	Minimal weight	$\pi_{\mathcal{J}}$	$O(n^3)$	$O(n^2)$	(53)
B – Partition function	Weighted count	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(n^3)$	$O(n^2)$	(32)
C – Base-pairing probabilities	Arc-traversal prob.	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(n^3)$	$O(n^2)$	(32)
D – Statistical sampling (k-samples)	Weighted random gen.	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(n^3 + k \cdot n \log n)$	$O(n^2)$	(11; 38)
E – Moments of energy (Mean, Var.)	Moments extraction	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(n^3)$	$O(n^2)$	(35)
F – m-th moment of additive features	Moments extraction	$e^{\frac{-\pi_{\mathcal{J}}}{RT}}$	$O(m^3.n^3)$	$O(m.n^2)$	–
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.

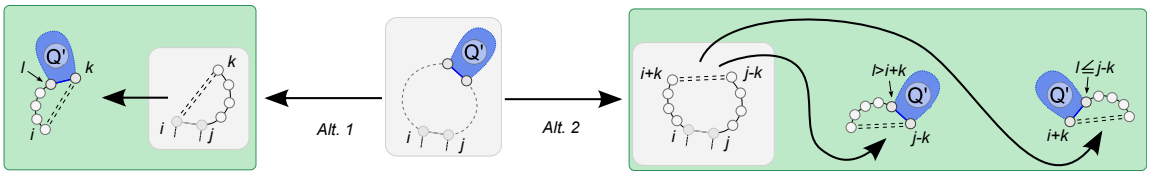

Fig. 3. Alternative exhaustive strategies for interior loops.

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 $K = 30$. 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**, as there exists for instance different ways to create an unpaired regions. Therefore we propose in Figure 4 an unambiguous decomposition for the same conformation space.

Previous results. In a previous work (44; 45), 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, which can be encoded by a well-parenthesized word p over two systems of parentheses $\{(x, \bar{x}), (y, \bar{y})\}$ and an unpaired character c, as

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

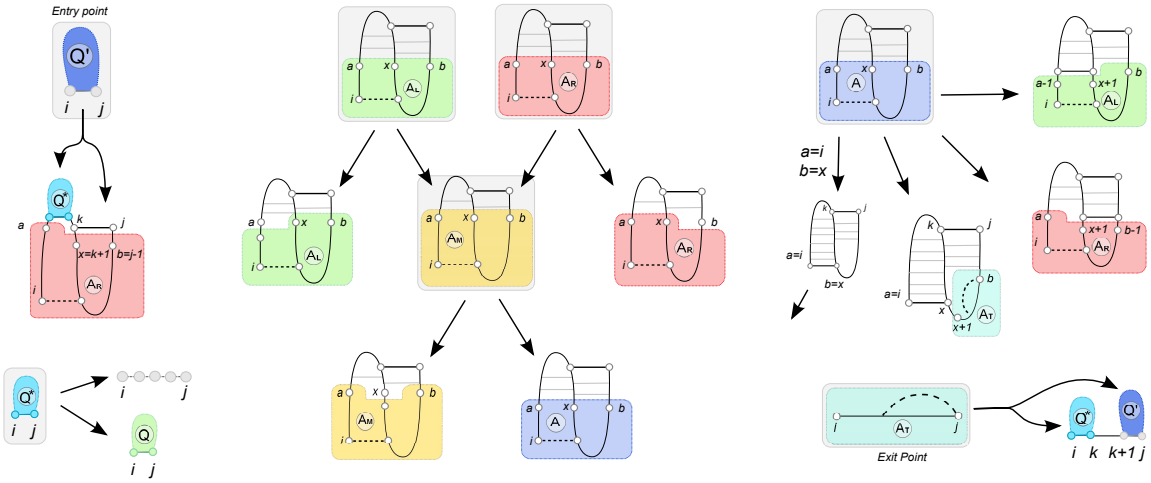

Fig. 4. An unambiguous decomposition for simple non-recursive pseudoknots that captures the Akutsu/Uemura class of pseudoknots. This decomposition yields $O(n^4)/\Theta(n^4)$ time/memory algorithms for partially recursive pseudoknots and can be extended to include recursive pseudoknots and/or Turner 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 $xp'y\bar{x}p''\bar{y}$ which can be generated by applying the initial case ($Q \to A_L \to A_M \to A \rightsquigarrow p'\dots y\dots\bar{y}$) followed directly by the terminal case ($A \to A_T \rightsquigarrow x\,p'\,y\,\bar{x}\,p''\,\bar{y}$). Moreover through a sequence $A \to A_R \to A_M \to A$, one adds an outermost edge around the right part $y\dots\bar{y}$. So through m iterations of the sequence the decomposition generates any structure $y^{m_1}\dots\bar{y}^{m_1}$. Similarly through a sequence $A \to A_L \to A_M \to A$ one adds an outermost edge around the left part $x\dots\bar{x}$, and after n_1 iterations any structure $x^{n_1}\dots\bar{x}^{n_1}$ 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

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

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 x of a leftward base pair (resp. after closure $\bar{x}$ of a leftward base pair, after the opening y of a right base pair and before the closure $\bar{y}$ 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:

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

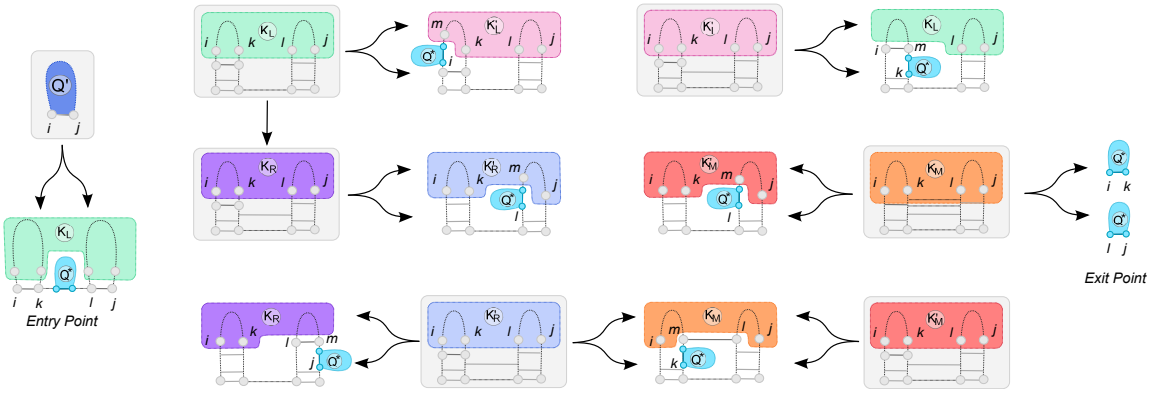

Fig. 5. Unambiguous decomposition of fully recursive kissing hairpins.

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* (7) in time complexity in $O(n^5)$, and in $O(n^3)/O(n^4)$ under restrictions by Theis *et al* (47). 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 (44), showing that any KH pseudoknot can be represented by a word p over three systems of parentheses $\{(x,\bar{x}),(y,\bar{y}),(z,\bar{z})\}$ (respectively denoting leftmost, central and rightmost helices) such that:

$$p = (xS)^n\,(yS)^m\,(\bar{x}S)^n\,(zS)^k\,(\bar{y}S)^m\,(\bar{z}S)^{k-1}\,\bar{z}. \tag{14}$$

Completeness. First let us remark that the *minimal* conformation generated by the decomposition is $K_L \to K_R \to K'_R \to K_M \rightsquigarrow xSySx̄SzSȳSz̄$. 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 $(K_L \to K'_L \to K_L)^{n-1} \rightsquigarrow (Sx)^{n-1}\cdots(\bar{x}S)^{n-1}$, $(K'_R \to K_R \to K'_R)^{k-1} \rightsquigarrow (zS)^{k-1}\cdots(\bar{z}S)^{k-1}$ and $(K_M \to K'_M \to K_M)^m \rightsquigarrow (yS)^{m-1}\cdots(S\bar{y})^{m-1}$ in the minimal word above, and produce any conformation denoted by

$$x(Sx)^{n-1}S(yS)^{m-1}yS(\bar{x}S)^{n-1}\bar{x}SzS(zS)^{k-1}\bar{y}(S\bar{y})^{m-1}S(\bar{z}S)^{k-1}\bar{z}$$

where one recognizes the language of Equation 14 upon simple refactorization.
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:

$$\begin{aligned}
&& K(z) &= z^4 K_L(z) S(z) && \\
K_L(z) &= S(z)K'_L(z) + K_R(z) & K'_L(z) &= z^2 K_L(z) S(z) & K_M(z) &= K'_M(z)S(z) + S(z)^2 \\
K'_M(z) &= z^2 K_M(z) S(z) & K_R(z) &= K'_R(z) S(z) & K'_R(z) &= z^2 K_R(z) S(z) + z^2 K_M(z) S(z)
\end{aligned}$$

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	$e^{\frac{-\pi_{bp}}{RT}}$	$O(n^4)$	$O(n^4)$	(5; 6) 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	$e^{\frac{-\pi_{bp}}{RT}}$	$O(m^3.n^4)$	$O(m.n^4)$	–
Fully recursive Kissing Hairpins					
A – Energy minimization	Minimal weight	$\pi_{\mathcal{F}}$	$O(n^5)$	$O(n^4)$	(7)
B – Partition function	Weighted count	$e^{\frac{-\pi_{\mathcal{F}}}{RT}}$	$O(n^5)$	$O(n^4)$	–
C – Base-pairing probabilities	Arc-traversal prob.	$e^{\frac{-\pi_{\mathcal{F}}}{RT}}$	$O(n^5)$	$O(n^4)$	–
D – Statistical sampling (k-samples)	Weighted rand. gen.	$e^{\frac{-\pi_{\mathcal{F}}}{RT}}$	$O(n^5 + k \cdot n \log n)$	$O(n^4)$	–
E – Moments of energy (Mean, Var.)	Moments extraction	$e^{\frac{-\pi_{\mathcal{F}}}{RT}}$	$O(n^5)$	$O(n^4)$	–
F – m-th moment of additive features	Moments extraction	$e^{\frac{-\pi_{\mathcal{F}}}{RT}}$	$O(m^3.n^5)$	$O(m.n^4)$	–

Table 2. Summary of ensemble based algorithms on simple pseudoknots and kissing hairpins. π_{bp} stands for the simple Nussinov-Jacobson energy model, and $\pi_{\mathcal{F}}$ 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* (35) proposed an $\mathcal{O}(m^2 \cdot n^3)$ 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-uplet 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 traversal/generation.

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 (15), 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.

There exists many perspectives to our contribution. First the principles and algorithms described here could easily be implemented as a general compiler tools for Forward-Hypergraph algorithms. Such a compiler could be coupled with helper tools expanding hypergraphs from succinct descriptions. Some of the candidates for such descriptions are context-free grammars (related to ADP (18)), or Matthias Möhl's split types (34). 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 (37)), those capturing RNA-RNA interactions (2; 23), those offering simultaneous alignment and folding (Sankoff's algorithm (43)) or performing mutations on the sequence (50). Finally hypergraph algorithms are not necessarily limited to dynamic-programming, and algorithmic developments could be proposed to address some of the current algorithmic issues in RNA (inverse folding (3), kinetics (46)) for which no exact polynomial algorithms are currently known. 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(49; 25; 41), 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.

Bibliography

[1] Akutsu, T.: Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. Discrete Appl. Math. 104(1-3), 45–62 (2000)

[2] Alkan, C., Karakoç, E., Nadeau, J.H., Sahinalp, S.C., Zhang, K.: RNA-RNA Interaction Prediction and Antisense RNA Target Search. In: Proceedings of RECOMB'05 (2005)

[3] Andronescu, M., Fejes, A.P., Hutter, F., Hoos, H.H., Condon, A.: A New Algorithm for RNA Secondary Structure Design. J Mol Biol 336(3), 607–624 (2004)

[4] Bekaert, M., Bidou, L., Denise, A., Duchateau-Nguyen, G., Forest, J., Froidevaux, C., Hatin, I., Rousset, J., Termier, M.: Towards a computational model for −1 eukaryotic frameshifting sites. Bioinformatics 19, 327–335 (2003)

[5] Cao, S., Chen, S.J.: Predicting RNA pseudoknot folding thermodynamics. Nucleic Acids Res 34(9), 2634–2652 (2006)

[6] Cao, S., Chen, S.J.: Predicting structured and stabilities for H-type pseudoknots with interhelix loop. RNA 15, 696–706 (2009)

[7] Chen, H.L., Condon, A., Jabbari, H.: 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)

[8] Condon, A., Davy, B., Rastegari, B., Zhao, S., Tarrant, F.: Classifying RNA pseudoknotted structures. Theoretical Computer Science 320(1), 35–50 (2004)

[9] Denise, A., Ponty, Y., Termier, M.: Controlled non uniform random generation of decomposable structures. Theoretical Computer Science 411(40-42), 3527–3552 (September 2010)

[10] Ding, Y., Chan, C.Y., Lawrence, C.E.: RNA secondary structure prediction by centroids in a boltzmann weighted ensemble. RNA 11, 1157–1166 (2005)

[11] Ding, Y., Lawrence, E.: A statistical sampling algorithm for RNA secondary structure prediction. Nucleic Acids Res 31(24), 7280–7301 (2003)

[12] Dirks, R., Pierce, N.: A partition function algorithm for nucleic acid secondary structure including pseudoknots. J Comput Chem 24, 1664–1677 (2003)

[13] Do, C.B., Woods, D.A., Batzoglou, S.: CONTRAfold: RNA secondary structure prediction without physics-based models. Bioinformatics 22(14), e90–e98 (Jul 2006)

[14] Ferrè, F., Ponty, Y., Lorenz, W.A., Clote, P.: 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)

[15] Finkelstein, A.V., Roytberg, M.A.: Computation of biopolymers: a general approach to different problems. Biosystems 30(1-3), 1–19 (1993)

[16] Flajolet, P., Zimmermann, P., Van Cutsem, B.: 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

[17] Flajolet, P.: Analytic models and ambiguity of context-free languages. Theoretical Computer Science 49, 283–309 (1987)

[18] Giegerich, R.: A systematic approach to dynamic programming in bioinformatics. Bioinformatics 16(8), 665–677 (Aug 2000)

[19] Hamada, M., Kiryu, H., Sato, K., Mituyama, T., Asai, K.: Prediction of RNA secondary structure using generalized centroid estimators. Bioinformatics 25(4), 465–473 (Feb 2009)

[20] Harmanci, A.O., Sharma, G., Mathews, D.H.: Stochastic sampling of the rna structural alignment space. Nucleic Acids Res 37(12), 4063–4075 (Jul 2009)

[21] Hofacker, I.L.: Vienna RNA secondary structure server. Nucleic Acids Res 31(13), 3429–3431 (Jul 2003)

[22] Huang, F.W.D., Peng, W.W.J., Reidys, C.M.: Folding 3-noncrossing rna pseudoknot structures. J Comput Biol 16(11), 1549–1575 (Nov 2009)

[23] Huang, F.W.D., Qin, J., Reidys, C.M., Stadler, P.F.: Target prediction and a statistical sampling algorithm for RNA-RNA interaction. Bioinformatics 26(2), 175–181 (Jan 2010), `http://dx.doi.org/10.1093/bioinformatics/btp635`

[24] Lefebvre, F.: A grammar-based unification of several alignment and folding algorithms. In: Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology. pp. 143–154. AAAI Press (1996)

[25] Lescoute, A., Westhof, E.: Topology of three-way junctions in folded RNAs. RNA 12(1), 83–93 (2006)

[26] Lorenz, W., Ponty, Y., Clote, P.: Asymptotics of RNA shapes. Journal of Computational Biology 15(1), 31–63 (Jan–Feb 2008)

[27] Lyngsø, R.B., Pedersen, C.N.S.: RNA pseudoknot prediction in energy-based models. Journal of Computational Biology 7(3-4), 409–427 (2000)

[28] Markham, N.R.: Algorithms and software for nucleic acid sequences. Ph.D. thesis, Faculty of Rensselaer Polytechnic Institute (2006)

[29] Markham, N.R., Zuker, M.: UNAFold: software for nucleic acid folding and hybridization. Methods Mol Biol 453, 3–31 (2008)

[30] Mathews, D.H.: Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. RNA 10(8), 1178–1190 (2004)

[31] Mathews, D., Sabina, J., Zuker, M., Turner, D.: Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. J Mol Biol 288, 911–940 (1999)

[32] McCaskill, J.: The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Biopolymers 29, 1105–1119 (1990)

[33] Mückstein, U., Hofacker, I.L., Stadler, P.F.: Stochastic pairwise alignments. Bioinformatics 18 Suppl 2, S153–S160 (2002)

[34] Möhl, M., Will, S., Backofen, R.: Lifting prediction to alignment of rna pseudoknots. J Comput Biol 17(3), 429–442 (Mar 2010), `http://dx.doi.org/10.1089/cmb.2009.0168`

[35] Miklós, I., Meyer, I.M., Nagy, B.: Moments of the boltzmann distribution for RNA secondary structures. Bull Math Biol 67(5), 1031–1047 (Sep 2005)

[36] Nussinov, R., Jacobson, A.B.: Fast algorithm for predicting the secondary structure of single stranded RNA. Proc. Natl. Acad. Sci. U. S. A. 77(11), 6309–6313 (1980)

[37] Parisien, M., Major, F.: The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. Nature 452(7183), 51–55 (2008)

[38] Ponty, Y.: 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)

[39] Reeder, J., Giegerich, R.: Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. BMC Bioinformatics 5, 104 (2004)

[40] Reeder, J., Steffen, P., Giegerich, R.: Effective ambiguity checking in biosequence analysis. BMC Bioinformatics 6, 153 (2005), `http://dx.doi.org/10.1186/1471-2105-6-153`

[41] Reidys, C.M., Huang, F.W.D., Andersen, J.E., Penner, R.C., Stadler, P.F., Nebel, M.E.: Topology and prediction of rna pseudoknots. Bioinformatics 27(8), 1076–1085 (Apr 2011)

[42] Rivas, E., Eddy, S.: A dynamic programming algorithm for RNA structure prediction including pseudoknots. J Mol Biol 285, 2053–2068 (1999)
[43] Sankoff, D.: Simultaneous solution of the rna folding, alignment and protosequence problems. SIAM J Appl Math 45, 810–825 (1985)
[44] Saule, C.: Modèles combinatoires des structures d'ARN avec ou sans pseudonœuds, application à la comparaison de structures. Ph.D. thesis, Université Paris Sud, Ecole doctorale informatique. (December 2010)
[45] Saule, C., Régnier, M., Steyaert, J.M., Denise, A.: Counting RNA pseudoknotted structures. Journal of Computational Biology (To appear)
[46] Thachuk, C., Manuch, J., Rafiey, A., Mathieson, L.A., Stacho, L., Condon, A.: An algorithm for the energy barrier problem without pseudoknots and temporary arcs. Pac Symp Biocomput pp. 108–119 (2010)
[47] Theis, C., Janssen, S., Giegerich, R.: Prediction of rna secondary structure including kissing hairpin motifs. In: Proceedings of WABI 2010. pp. 52–64 (2010)
[48] Tinoco, I., Borer, P.N., Dengler, B., Levin, M.D., Uhlenbeck, O.C., Crothers, D.M., Bralla, J.: Improved estimation of secondary structure in ribonucleic acids. Nat New Biol 246(150), 40–41 (Nov 1973)
[49] Vernizzi, G., Ribeca, P., Orland, H., Zee, A.: Topology of pseudoknotted homopolymers. Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) 73(3), 031902 (2006)
[50] Waldispühl, J., Devadas, S., Berger, B., Clote, P.: Efficient algorithms for probing the RNA mutation landscape. PLoS Comput Biol 4(8), e1000124 (2008), http://dx.doi.org/10.1371/journal.pcbi.1000124
[51] Waterman, M.S.: Secondary structure of single stranded nucleic acids. Advances in Mathematics Supplementary Studies 1(1), 167–212 (1978)
[52] Wilf, H.S.: A unified setting for sequencing, ranking, and selection algorithms for combinatorial objects. Advances in Mathematics 24, 281–291 (1977)
[53] Zuker, M., Stiegler, P.: Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res 9, 133–148 (1981)

Proof of Theorem 1

Lemma 1. *Let $\{x_{i,j}\}_{\substack{1\leq i\leq N\\1\leq j\leq M}}$ be a set of $N\times M$ real-valued coefficients, and $(a_i)_{1\leq i\leq N}\in\mathbb{N}^N$ be a tuple of integer exponents. Furthermore let us remind the multinomial notation*

$$\binom{a}{b_1,b_2,\cdots,b_M}=\frac{a!}{b_1!b_2!\cdots b_M!}.$$

Then the following identity holds

$$\prod_{i=1}^{N}\left(\sum_{j=1}^{M}x_{i,j}\right)^{a_i}=\sum_{\substack{\mathbf{b}_1:=(b_{1,1},\cdots,b_{1,M})\\\cdots\\\mathbf{b}_N:=(b_{N,1},\cdots,b_{N,M})\\ s.t.\ \sum_{j=1}^{M}b_{i,j}=a_i,\forall i\in[1,N]}}\prod_{i=1}^{N}\left(\binom{a_i}{b_{i,1},\cdots,b_{i,M}}\prod_{j=1}^{M}x_{i,j}^{b_{i,j}}\right)\tag{18}$$

Proof. To prove Equation 18, we are going to proceed by induction on N. For $N=1$, Equation 18 specializes into

$$(x_{1,1}+x_{1,2}+\cdots+x_{1,M})^{a_1}=\sum_{\substack{\mathbf{b}_1:=(b_{1,1},\cdots,b_{1,M})\\\text{s.t. }\sum_{j=1}^{M}b_{1,j}=a_1}}\binom{a_1}{b_{1,1},b_{1,2},\cdots,b_{1,M}}\prod_{j=1}^{M}x_{1,j}^{b_{1,j}},$$

in which one recognizes the multinomial coefficients formula.

Assume that Equation 18 holds for $N=K-1$. Then for $N=K$, one has

$$\prod_{i=1}^{K}\left(\sum_{j=1}^{M}x_{i,j}\right)^{a_i}=\prod_{i=1}^{K-1}\left(\sum_{j=1}^{M}x_{i,j}\right)^{a_i}\times\left(\sum_{j=1}^{M}x_{k,j}\right)^{a_k}=\left(\sum_{\substack{(\mathbf{b}_1,\cdots,\mathbf{b}_{K-1}),\\\sum_{j=1}^{M}b_{i,j}=a_i}}\prod_{i=1}^{K-1}\binom{a_i}{b_{i,1},\cdots,b_{i,M}}\prod_{j=1}^{M}x_{i,j}^{b_{i,j}}\right)\times\left(\sum_{j=1}^{M}x_{k,j}\right)^{a_k}.$$

The right-hand side term of the product can be developed using the multinomial coefficients formula, and it follows that

$$\prod_{i=1}^{K}\left(\sum_{j=1}^{M}x_{i,j}\right)^{a_i}=\left(\sum_{\substack{(\mathbf{b}_1,\cdots,\mathbf{b}_{K-1}),\\\sum_{j=1}^{M}b_{i,j}=a_i}}\prod_{i=1}^{K-1}\binom{a_i}{b_{i,1},\cdots,b_{i,M}}\prod_{j=1}^{M}x_{i,j}^{b_{i,j}}\right)\times\left(\sum_{\substack{\mathbf{b}_K\text{ s. t.}\\\sum_{j=1}^{M}b_{K,j}=a_K}}\binom{a_K}{b_{K,1},\cdots,b_{K,M}}\prod_{j=1}^{M}x_{K,j}^{b_{K,j}}\right).$$

Since the two terms of the highest-level product are mutually independent, we can treat the right-hand as a constant and inject it within the left-hand sum.

$$\prod_{i=1}^{K}\left(\sum_{j=1}^{M}x_{i,j}\right)^{a_i}=\sum_{\substack{(\mathbf{b}_1,\cdots,\mathbf{b}_{K-1}),\\\sum_{j=1}^{M}b_{i,j}=a_i}}\left(\left(\sum_{\substack{\mathbf{b}_K\text{ s. t.}\\\sum_{j=1}^{M}b_{K,j}=a_K}}\binom{a_K}{b_{K,1},\cdots,b_{K,M}}\prod_{j=1}^{M}x_{K,j}^{b_{K,j}}\right)\times\prod_{i=1}^{K-1}\binom{a_i}{b_{i,1},\cdots,b_{i,M}}\prod_{j=1}^{M}x_{i,j}^{b_{i,j}}\right)$$

This argument can be used to inject the right-hand-side products within the inner sum. It follows that

$$\begin{aligned}\prod_{i=1}^{K}\left(\sum_{j=1}^{M}x_{i,j}\right)^{a_i}&=\sum_{\substack{(\mathbf{b}_1,\cdots,\mathbf{b}_{K-1}),\\\sum_{j=1}^{M}b_{i,j}=a_i}}\sum_{\substack{\mathbf{b}_K\text{ s. t.}\\\sum_{j=1}^{M}b_{K,j}=a_K}}\left(\binom{a_K}{b_{K,1},\cdots,b_{K,M}}\prod_{j=1}^{M}x_{K,j}^{b_{K,j}}\prod_{i=1}^{K-1}\binom{a_i}{b_{i,1},\cdots,b_{i,M}}\prod_{j=1}^{M}x_{i,j}^{b_{i,j}}\right)\\&=\sum_{\substack{(\mathbf{b}_1,\cdots,\mathbf{b}_{K-1}),\\\sum_{j=1}^{M}b_{i,j}=a_i}}\sum_{\substack{\mathbf{b}_K\text{ s. t.}\\\sum_{j=1}^{M}b_{K,j}=a_K}}\left(\prod_{i=1}^{K}\binom{a_i}{b_{i,1},\cdots,b_{i,M}}\prod_{j=1}^{M}x_{i,j}^{b_{i,j}}\right)\\&=\sum_{\substack{(\mathbf{b}_1,\cdots,\mathbf{b}_K),\\\sum_{j=1}^{M}b_{i,j}=a_i}}\left(\prod_{i=1}^{K}\binom{a_i}{b_{i,1},\cdots,b_{i,M}}\prod_{j=1}^{M}x_{i,j}^{b_{i,j}}\right).\end{aligned}$$

So the validity of the initial statement for $N = K-1$ implies its validity for $N = K$ which, in combination with its initial validity at $N = 0$, implies the universal validity of the identity. □

Theorem 2. *Let* $\mathbf{m} := (m_1, \cdots, m_k)$ *be a* k*-uplet of natural integers.*
Then the pseudo-moment $c_s^{\mathbf{m}} := \mathbb{E}[X_1^{m_1} \cdots X_k^{m_k} \mid s] \cdot w_s$ *can be recursively computed through*

$$c_s^{\mathbf{m}} = \sum_{e=(s\to\mathbf{t})} \pi(e) \cdot \sum_{\substack{\mathbf{m}', \left(\mathbf{m}''_1, \cdots, \mathbf{m}''_{|t|}\right) \\ s.\,t.\ \mathbf{m}' + \sum_j \mathbf{m}''_j = \mathbf{m}}} \prod_{i=1}^{k} \binom{m_i}{m'_i, m''_{1,i}, \cdots, m''_{|t|,i}} \cdot \alpha_i(e)^{m'_i} \cdot \prod_{i=1}^{|t|} c_{t_i}^{\mathbf{m}''_i}. \tag{19}$$

Proof. Since the F-graph is acyclic, one can use a bottom-up induction, starting from vertices s^* at height 0, i.e. having only terminal edges. For such nodes, Equation 19 simplifies into

$$c_{s^*}^{\mathbf{m}} = \sum_{e=(s^*\to\varnothing)} \pi(e) \cdot \prod_{i=1}^{k} \binom{m_i}{m_i} \cdot \alpha_i(e)^{m_i} = \sum_{e=(s^*\to\varnothing)} \pi(e) \cdot \prod_{i=1}^{k} \alpha_i(e)^{m_i} = \sum_{p\in\mathscr{P}_{s^*}} \pi(p) \cdot \prod_{i=1}^{k} \alpha_i(p)^{m_i}$$

in which one recognizes the definition of the pseudo-moment.

Let us now assume that Equation 19 was able to compute the pseudo-moments for any outgoing vertex of s. It follows that, for any such vertex t_i and any k-uplet of integers $\mathbf{m}''_i$, one has

$$c_{t_i}^{\mathbf{m}''_i} = \sum_{p_i\in\mathscr{P}_{t_i}} \pi(p_i) \cdot \prod_{j=1}^{k} \alpha_j(p_i)^{m''_j}$$

$$\prod_{i=1}^{|t|} c_{t_i}^{\mathbf{m}''_i} = \prod_{i=1}^{|t|} \left(\sum_{p_i\in\mathscr{P}_{t_i}} \pi(p_i) \cdot \prod_{j=1}^{k} \alpha_j(p_i)^{m''_j} \right) = \sum_{\substack{(p_1,\cdots,p_{|t|}) \\ \text{s.t. } p_i\in\mathscr{P}_{t_i}}} \left(\left(\prod_{i=1}^{|t|} \pi(p_i) \right) \times \prod_{i=1}^{|t|} \prod_{j=1}^{k} \alpha_j(p_i)^{m''_{i,j}} \right).$$

Now applying the formula of Equation 19 to a vertex s, one obtains

$$c_s^{\mathbf{m}} = \sum_{e=(s\to\mathbf{t})} \pi(e) \cdot \sum_{\substack{\mathbf{m}', \left(\mathbf{m}''_1, \cdots, \mathbf{m}''_{|t|}\right) \\ \text{s. t. } \mathbf{m}' + \sum_j \mathbf{m}''_j = \mathbf{m}}} \left(\prod_{i=1}^{k} \binom{m_i}{m'_i, m''_{1,i}, \cdots, m''_{|t|,i}} \cdot \alpha_i(e)^{m'_i} \right) \sum_{\substack{(p_1,\cdots,p_{|t|}) \\ \text{s.t. } p_i\in\mathscr{P}_{t_i}}} \left(\left(\prod_{i=1}^{|t|} \pi(p_i) \right) \times \prod_{i=1}^{|t|} \prod_{j=1}^{k} \alpha_j(p_i)^{m''_{i,j}} \right)$$

$$= \sum_{e=(s\to\mathbf{t})} \pi(e) \cdot \sum_{\substack{\mathbf{m}', \left(\mathbf{m}''_1, \cdots, \mathbf{m}''_{|t|}\right) \\ \text{s. t. } \mathbf{m}' + \sum_j \mathbf{m}''_j = \mathbf{m}}} \sum_{\substack{(p_1,\cdots,p_{|t|}) \\ \text{s.t. } p_i\in\mathscr{P}_{t_i}}} \left(\left(\prod_{i=1}^{k} \binom{m_i}{m'_i, m''_{1,i}, \cdots, m''_{|t|,i}} \cdot \alpha_i(e)^{m'_i} \right) \left(\prod_{i=1}^{|t|} \pi(p_i) \right) \times \prod_{i=1}^{|t|} \prod_{j=1}^{k} \alpha_j(p_i)^{m''_{i,j}} \right).$$

Since the sums on $(\mathbf{m}', \mathbf{m}'')$ and $(p_1, \cdots, p_{|t|})$ are mutually independent, their order of summation can be exchanged. Furthermore the $\pi(\cdot)$ terms do not depend on $(\mathbf{m}', \mathbf{m}'')$ and $(p_1, \cdots, p_{|t|})$, so they can be moved out of the summations.

$$c_s^{\mathbf{m}} = \sum_{e=(s\to\mathbf{t})} \pi(e) \cdot \sum_{\substack{(p_1,\cdots,p_{|t|}) \\ \text{s.t. } p_i\in\mathscr{P}_{t_i}}} \sum_{\substack{\mathbf{m}', \left(\mathbf{m}''_1, \cdots, \mathbf{m}''_{|t|}\right) \\ \text{s. t. } \mathbf{m}' + \sum_j \mathbf{m}''_j = \mathbf{m}}} \left(\left(\prod_{i=1}^{k} \binom{m_i}{m'_i, m''_{1,i}, \cdots, m''_{|t|,i}} \cdot \alpha_i(e)^{m'_i} \right) \left(\prod_{i=1}^{|t|} \pi(p_i) \right) \times \prod_{i=1}^{|t|} \prod_{j=1}^{k} \alpha_j(p_i)^{m''_{i,j}} \right)$$

$$= \sum_{e=(s\to\mathbf{t})} \cdot \sum_{\substack{(p_1,\cdots,p_{|t|}) \\ \text{s.t. } p_i\in\mathscr{P}_{t_i}}} \pi(e) \left(\prod_{i=1}^{|t|} \pi(p_i) \right) \sum_{\substack{\mathbf{m}', \left(\mathbf{m}''_1, \cdots, \mathbf{m}''_{|t|}\right) \\ \text{s. t. } \mathbf{m}' + \sum_j \mathbf{m}''_j = \mathbf{m}}} \left(\left(\prod_{i=1}^{k} \binom{m_i}{m'_i, m''_{1,i}, \cdots, m''_{|t|,i}} \cdot \alpha_i(e)^{m'_i} \right) \times \prod_{i=1}^{|t|} \prod_{j=1}^{k} \alpha_j(p_i)^{m''_{i,j}} \right)$$

$$= \sum_{e=(s\to\mathbf{t})} \sum_{\substack{(p_1,\cdots,p_{|t|}) \\ \text{s.t. } p_i\in\mathscr{P}_{t_i}}} \pi(e) \left(\prod_{i=1}^{|t|} \pi(p_i) \right) \sum_{\substack{\mathbf{m}', \left(\mathbf{m}''_1, \cdots, \mathbf{m}''_{|t|}\right) \\ \text{s. t. } \mathbf{m}' + \sum_j \mathbf{m}''_j = \mathbf{m}}} \prod_{i=1}^{k} \left(\left(\binom{m_i}{m'_i, m''_{1,i}, \cdots, m''_{|t|,i}} \times \alpha_i(e)^{m'_i} \times \prod_{j=1}^{|t|} \alpha_i(p_j)^{m''_{j,i}} \right) \right).$$

From a direct application of Lemma 1, we have

$$\prod_{i=1}^{k}\left(\binom{m_i}{m'_i, m''_{1,i}, \cdots, m''_{|t|,i}} \times \alpha_i(e)^{m'_i} \times \prod_{j=1}^{|t|} \alpha_i(p_j)^{m''_{j,i}}\right) = \prod_{i=1}^{k}\left(\alpha_i(e) + \sum_{j=1}^{|t|} \alpha_i(p_j)\right)^{m_i}$$

and it follows that

$$c_s^{\mathbf{m}} = \sum_{e=(s\to\mathbf{t})} \sum_{\substack{(p_1,\cdots,p_{|t|}) \\ \text{s.t. } p_i\in\mathscr{P}_{t_i}}} \left(\pi(e)\prod_{i=1}^{|t|}\pi(p_i)\right)\prod_{i=1}^{k}\left(\alpha_i(e) + \sum_{j=1}^{|t|}\alpha_i(p_j)\right)^{m_i}.$$

At this point, let us make a few observations to conclude. Consider an F-path p, starting with an F-arc $e = (s \to \mathbf{t})$, followed by $|t|$ F-paths $(p_1, \cdots, p_{|t|})$ such that $p_i \in \mathscr{P}_{t_i}, \forall\, i \in [1, |t|]$:

- From the multiplicative definition of the weight, we know that $\pi(p) = \pi(e)\prod_{i=1}^{|t|}\pi(p_i)$.
- From the additive definition of features, we know that $\alpha_j(p) = \alpha_j(e) + \sum_{i=1}^{|t|}\alpha_j(p_i)$.
- Any path originating from s can be uniquely decomposed as an edge followed by a tuple of paths originating from its outgoing edges.

It follows that

$$c_s^{\mathbf{m}} = \sum_{p\in\mathscr{P}_s} \pi(p) \times \prod_{i=1}^{k}\alpha_i(p)^{m_i} = \mathbb{E}[X_1^{m_1}\cdots X_k^{m_k} \mid s]\cdot w_s$$

and that the validity of the recurrence carries from children vertices to parents which, in conjunction with the acyclicity of the F-graph proves the correctness of our claim. □

Theorem 3 (Completeness). *Assume that $\mathscr{D}$ is unambiguous, then $\mathscr{D}$ is complete iff $S(z) = S_{\mathscr{D}}(z)$.*

Proof. First let us point out that the unambiguity of $\mathscr{D}$ implies that $\mathscr{S}_{\mathscr{D}}$, initially defined as a multiset, is in fact a set. Let us then remind that a decomposition is complete iff $\mathscr{S}_{\mathscr{D}} = \mathscr{S}$. Let us denote by E_n the subset of a set E such that each object $e \in E_n$ has size $|e| = n$, and use this notation to rewrite the generating functions as $S(z) = \sum_{n\geq 0}|\mathscr{S}_n|\cdot z^n$ and $S_{\mathscr{D}}(z) = \sum_{n\geq 0}|\mathscr{S}_{\mathscr{D},n}|\cdot z^n$. Consequently $S(z) \neq S_{\mathscr{D}}(z)$ implies that there exists some n_0 such that $|\mathscr{S}_{\mathscr{D},n_0}| \neq |\mathscr{S}_{n_0}|$ and therefore $\mathscr{S}_{\mathscr{D}} \neq \mathscr{S}$ which proves the forward implication. Conversely $S(z) = S_{\mathscr{D}}(z)$ implies that for all positive value of n the equality $|\mathscr{S}_n| = |\mathscr{S}_{\mathscr{D},n}|$ holds. Since $\mathscr{S}_{\mathscr{D}} \subset \mathscr{S}$ then we have $\mathscr{S}_n \subset \mathscr{S}_{\mathscr{D},n}$ and the equality of cardinality, in conjunction with the unambiguity, implies that $\mathscr{S}_n = \mathscr{S}_{\mathscr{D},n}$. We conclude by pointing out that the size function induces a partition of $\mathscr{S}_{\mathscr{D}}$ and $\mathscr{S}$ and therefore $\mathscr{S}_{\mathscr{D}} = \mathscr{S}$. □