RNA Folding

Yann Ponty, Vladimir Reinharz

RNA Folding

Yann **Ponty**[1], Vladimir **Reinharz**[2]
[1] LIX UMR 7161, Polytechnic School, Polytechnic Institute of Paris, France
[2] Department of Computer Science, University of Quebec at Montréal, Canada

1 Introduction

Ribonucleic Acids (RNA) are molecules that are essential to any living organism. They are composed of nucleotides named Adenine, Cytosine, Guanine and Uracile, and usually represented as sequences over an $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$ alphabet. Such sequences differ slightly from DNA, itself encoded on an $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ alphabet, since Thymines are replaced by Uracile upon transcription. Functional RNA molecules feature a wide variety of sizes, ranging from 25 nucleotides (nts) for the aptly named micro RNAs, to dozen of thousands for viruses using RNA as its primary genomic material. For instance, HIV-1 encodes its function in a single-stranded RNA of approximately 9,500 nts while the genome of SARS-CoV 2, responsible for the COVID-19 pandemic, consists of an RNA molecule of more than 30,000 nts.

This length diversity reflects a wide functional diversity. Acting as mediators, messenger RNAs consist of slices of the genetic information contained in the DNA, and are used as a template for the synthesis of proteins. RNAs also play an integral part in the ribosome, a large multi-molecular assembly which translates messenger RNAs into proteins. They also regulate gene expression at a quantitative level, for example through the process of RNA interference, where small single-stranded RNAs bind to messenger RNAs, preventing the ribosome from binding to then, and thus inhibiting the synthesis of associated proteins.

Far from being exhaustive, this list of functions is also constantly growing, mirroring our ever-increasing discovery of novel RNAs. The RFAM database [Kalvari et al., 2017], which lists and organizes documented RNAs into functional families, has been experiencing a constant growth since its creation in 2002. As of 2022, RFAM lists more than 4000 functional families, as shown in Figure 1.

1.1 RNA folding

Unlike DNA, RNA is synthesized as a single molecule and does not necessarily adopt a regular double-helix structure like DNA. On the contrary, RNA is initially **single-stranded** during its synthesis, and folds back on itself through a process that is subject to nanoscale fluctuations. It is stabilized in some of its conformations, or structures, by

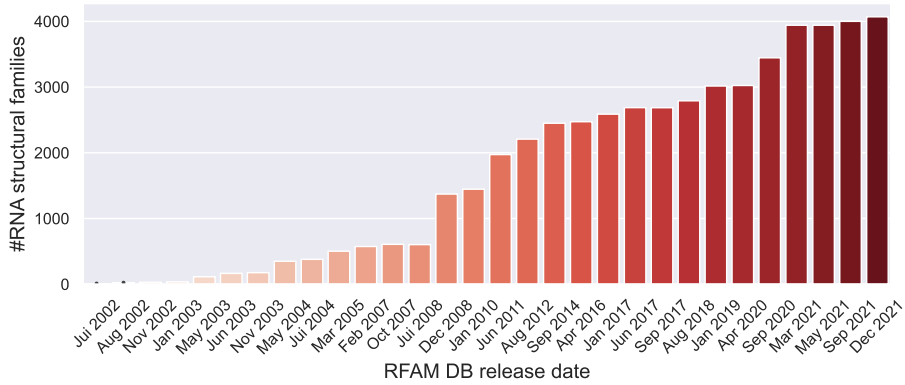


Figure 1: Evolution of the number of functional RNA families, indexed in the RFAM database [Kalvari et al., 2017]. The strong increase in the number of families in 2009 coincides with the democratization of the RNA-seq technology [ENCODE et al., 2007].

the formation of **base pairs**, connecting some of those nucleotides through hydrogen bonds.

Amongst non-coding RNAs, which act directly as RNAs and not through translation into proteins, a well-defined structure often constitutes an essential determinant of function. Consequently, across many functional families, the adoption of a precise structure is more conserved throughout evolution than a certain nucleotide sequence. Predicting the functional structure of an RNA therefore represents a necessary first step to understand its mode(s) of action, and to figure out its role within the broader context of biological systems.

1.1.1 Paradigms for folding prediction

From a perspective inspired by statistical physics, illustrated by Figure 2, RNA is initially transcribed in an essentially unstructured form (A). It then fluctuates in a stochastic manner, moving between its different states, *aka* **structures** or **conformations**. The system eventually reaches **thermodynamic equilibrium**, where the probability of observing an RNA in a given conformation ceases to evolve with time.

At the thermodynamic equilibrium, the set of structures possibly adopted by an RNA ω follows a **Boltzmann distribution**, such that a structure S is observed with probability

$$\mathbb{P}(S \mid \omega) = \frac{e^{-E(S)/RT}}{\mathcal{Z}}$$

where R is the Boltzmann constant, $E(S)$ is the **free energy** of S, T the temperature, and $\mathcal{Z}$ the **partition function**, a crucial quantity which can be seen here as a

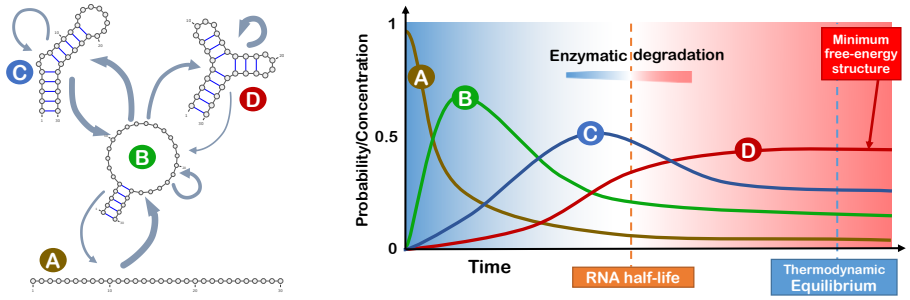


Figure 2: Main paradigms for RNA folding prediction. The folding process can be abstracted as a continuous time Markov process (left), whose states (A,B,C & D) represent the main conformations adopted by a toy RNA. Left for sufficient time, the process converge to a thermodynamic equilibrium (right) where the probability of observing RNA in a given structure depends only on its free-energy. The most probable structure is then the one with minimum free-energy, but others may sometimes dominate the thermodynamic equilibrium.

renormalization constant. This distribution maximizes the entropy given the average, measurable energy of the system.

This distribution motivates a focus, shared by many predictive approaches, on the **Minimum Free-Energy (MFE) structure**. Indeed, the MFE provably possesses the highest probability in the Boltzmann distribution, and thus represents the most likely structure to be observed by its potential partners in the cellular environment.

However, although maximal among structures, the MFE probability may be very small in absolute terms and, in the absence of evolutionary pressure, is even assumed to decrease exponentially with the length of the RNA. Rather than focus on a single poorly representative structure, some approaches will instead consider the expected properties of folding at **thermodynamic equilibrium**. For instance, in Figure 2 the outermost helix, involving both ends of the RNA, is present in structures B, C and D. It is therefore much more probably than the two apical hairpin loops, only found in the MFE structure D. Ensemble approaches, based on the explicit computation of the partition function and/or sampling techniques, make it possible to calculate these average properties, as well as representative structures (*eg* centroids of the dominant clusters).

More recent works focus on the **kinetics** of RNA, considering the properties of RNA folding before it reaches the thermodynamic equilibrium. Indeed, several phenomena (co-transcriptional folding [Lai et al., 2013], multi-stable RNAs [Findeiß et al., 2017]) exhibit a dependency on the initial distribution, refuting the hypothesis of convergence to an equilibrium. This out-of-equilibrium behavior can be explained by the limited life span of RNA, induced by an enzymatic degradation which prevents RNA from overcoming some **energy barriers** in time comparable to its **half-life**.

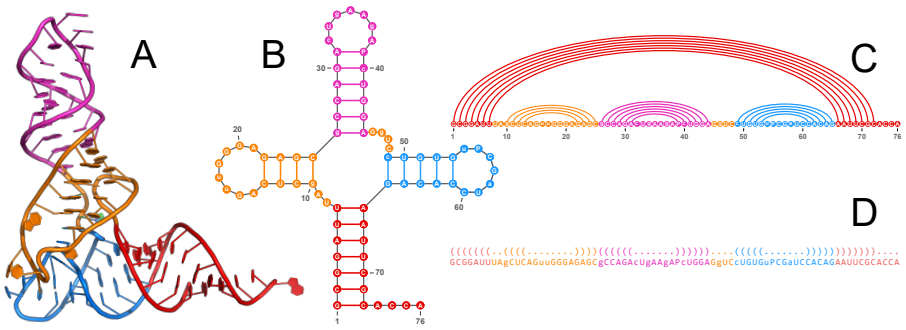


Figure 3: Representation of the secondary structure associated with the 3D folding of a transfer RNA (PDB 1EHZ, chain A).

Finally, evolution can help predict the functional structure of RNA, by postulating the existence of selection pressure constraining the structure of **homologous RNAs** to adopt the same function. From a reverse the point of view, a collection of homologous RNAs is likely to collectively adopt a common structure. This induces additional constraints, which supplement the energy model to inform the prediction of a shared structure.

1.2 Secondary structure

It is generally accepted that RNA adopts its functional structure through a folding process that is hierarchical in nature. RNA initially adopts a tree-like structure called the **secondary structure**, through a set of canonical pairings. In a second stage, RNA adopts a complex three-dimensional structure made possible by weaker stabilizing motifs.

Among these, we can find non-canonical bonds and complex topological patterns called **pseudoknots**, consisting of base pairs that are crossing when drawn in the upper half-plane. For this reason, structure prediction often starts with the critical guess of one (or more) candidate secondary structure(s). This initial prediction is then completed by additional crossing elements, and finally by the three-dimensional arrangement of these patterns.

Formally, a secondary structure is a set $S \subset [1, n]^2$ of base pairs $(i, j), 1 \leq i < j \leq n$, satisfying the following constraints:

1. Minimum distance θ: if $(i, j) \in S$, it then follows that $j - i > \theta$.
2. Monogamy: any position is involved in *at most* one pair of S.
3. Forbidden crossings: if $(i, j), (k, l) \in S$ such that $i < k$, it then follows that $i < k < l < j$ or $i < j < k < l$.

Under the aforementioned restrictions, the secondary structure can be represented in several forms, as illustrated in Figure 3. From a 3D RNA structure in the Protein Data Bank (PDB [Berman et al., 2000] – A), a secondary structure can be extracted, *e.g.*] using the `DSSR` program [Lu et al., 2015]. It can then be drawn without crossing as an outerplanar graph (B), or an arc-annotated sequence (C). Ultimately, a secondary structure can be represented very compactly using the **dot-bracket notation** (D), that is a sequence $t \in \{(,), \bullet\}^\star$ such that:

- There are as many opening and closing parentheses ($|t|_(= |t|_)$);
- For any prefix $t' \sqsubseteq t$, one has $|t'|_(\geq |t'|_)$.

In this setting, each opening parenthesis is unambiguously associated with a closing parenthesis, representing a base pair. The positions presenting a character $\bullet$ are then left free of interactions, or **unpaired**.

We finally describe the set of candidate structures, possibly adopted by an RNA sequence $\omega \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}^n$. A secondary structure S is **compatible** with ω if any base pair $(i, j) \in S$ is **canonical**, that is either a **Watson-Crick** (`G-C` or `A-U`) or **Wobble** base pair. More formally:

$$(\omega_i, \omega_j) \in \{(\mathtt{G}, \mathtt{C}), (\mathtt{C}, \mathtt{G}), (\mathtt{A}, \mathtt{U}), (\mathtt{U}, \mathtt{A}), (\mathtt{G}, \mathtt{U}), (\mathtt{U}, \mathtt{G})\}\,.$$

The set of structures compatible with the RNA ω is denoted by $\mathcal{S}_\omega$ (or simply $\mathcal{S}$ when clear from the context), and $\mathcal{S}_{i,j}$ represents the set of secondary structures that are compatible with the region $[i, j]$ of ω.

1.2.1 Energy model and structure space decomposition

RNA stability is physically determined by its free energy, expressed in kcal.mol^{-1}. The lower the free energy, the more stable an RNA structure is. The free energy of a structure depends largely on its base pairs, and their interaction in the form of patterns stabilizing the RNA structure.

In order to illustrate the different algorithmic approaches available for structure prediction, we shall consider a **simple energy model**, defined additively over base pairs. More precisely, let S be a secondary structure for a sequence ω, one has:

$$E(S) := \sum_{(i,j)\in S} E^\omega_{i,j}$$

where $E^\omega_{i,j}$ is the energy difference associated with the creation of the pair (i, j).

An energy of -1 can be associated with canonical base pairs (`G-C`, `A-U`, and `G-U`), and energy of $+\infty$ can be associated to invalid base pair. In other words, in this simple model, minimizing the free-energy coincides with maximizing the number of canonical pairs. Alternatively, one might consider an energy model that fosters base pairs considered to be more stable (`G-C` $\to -3$, `A-U` $\to -2$), disadvantaging those that are more transient (`G-U` $\to -1$).

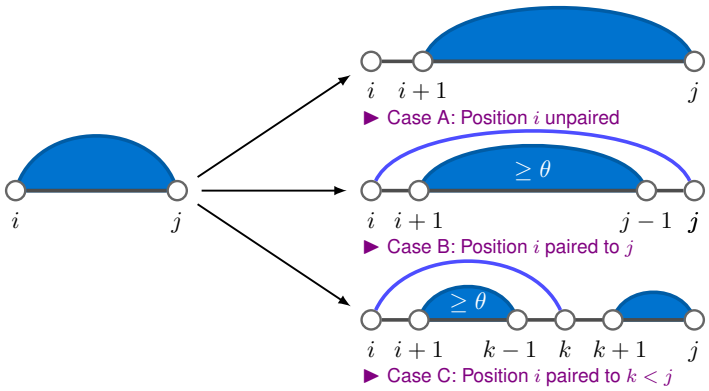


Figure 4: Decomposition of the space of secondary structures, possibly adopted by an RNA on a region $[i, j], 1 \leq i < j \leq n$, given a minimal base pair distance of θ nucleotides.

As shown in Figure 4, it is possible to decompose the secondary structures of $\mathcal{S}_{i,j}$ compatible with a region $[i, j]$ of a RNA ω. For this purpose, one considers the status of nucleotide i in a structure formed over $[i, j]$:

Case A Either i is unpaired, and followed by a secondary structure formed independently over the region $[i + 1, j]$;

Case B Or i is paired to a position j, $j - i > \theta$, and any secondary structure is formed over the region $[i + 1, j - 1]$;

Case C Or i is paired to a position $k < j$, $k - i > \theta$, and two structures are then formed in regions $[i + 1, k - 1]$ and $[k + 1, j]$. These latter are independent, due to pseudoknots being forbidden.

It can be shown that this decomposition is **complete**, *i.e.* any structure of $\mathcal{S}_{i,j}$ is generated/decomposed by one of the three cases above. Moreover, it is **unambiguous**, and any structure of $\mathcal{S}_{i,j}$ can be uniquely generated/decomposed through a recursive application of the three cases. Finally, it is **correct** for our simple energy model, as it explicitly scores base pairs, thus making it possible to capture our energy model.

2 Optimization for structure prediction

2.1 Computing the minimum free-energy (MFE) structure

The **Minimum Free-Energy structure** (MFE) represents the most stable structure among all of the structures adopted by a sequence. It is also the most probable struc-

ture at the thermodynamic equilibrium and, as such, represents a reasonable candidate while searching for the functional conformation for a given RNA.

Nonetheless, to determine this structure, one must overcome a combinatorial explosion of the number of secondary structures, asymptotically equivalent to $\sim 1.8^n$ [Zuker and Sankoff, 1984] on average for an RNA sequence of length n. The computation of a minimum energy structure thus represents a, potentially difficult, combinatorial optimization problem which can be defined as follows.

Free energy minimization

Input: Sequence $\omega \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}^+$, $|\omega| = n$.
Output: Secondary structure $S^\star$ such that

$$E(S^\star) = \min_{S \in \mathcal{S}} E(S)$$

A dynamic programming scheme for this problem is based on the decomposition introduced in Section 1.2.1. With this scheme, we are looking into the minimal energy $m_{i,j}$ accessible by folding the $[i, j]$ region within the input RNA. In any structure over $[i, j]$, and thus any minimum energy structure, only three options are possible for position i: Either i is free, and the structure minimum energy is found by optimizing the energy over $[i + 1, j]$; or i is paired to j, and an optimal folding forms over $[i + 1, j - 1]$; or position i is paired to some $k < j$, and two optimal and independent folds form over $[i + 1, k - 1]$ and $[k + 1, j]$.

It follows that, for a region $[i, j]$ such that $j - i > \theta$, the minimum free-energy $m_{i,j}$ over the region $[i, j]$ obeys:

$$m_{i,j} = \min \begin{cases} m_{i+1,j} & \blacktriangleright \text{ Case A: Pos. } i \text{ is free} \\ E^{\omega}_{i,j} + m_{i+1,j-1} & \blacktriangleright \text{ Case B: Pair } (i, j) \\ \min\limits_{k=i+\theta+1}^{j-1} E^{\omega}_{i,k} + m_{i,k-1} + m_{k+1,j} & \blacktriangleright \text{ Case C: Pair } (i, k), k > j \end{cases} \quad (1)$$

When $j - i \leq \theta$ one gets $m_{i,j} = 0$ because the region is then too small to contain a base pair.

We need to formulate an algorithm to efficiently compute this recurrence. Moreover, we also want to produce a minimum free-energy structure, and not just its energy, so we need to find an algorithm to reconstruct the chosen structure. Two algorithms are thus necessary, with quite similar structures:

- Algorithm 1 computes the minimum energy associated with each region $[i, j]$ in the sequence, as described in the system (1).
- Algorithm 2 backtracks, to reconstruct a minimum energy structure $S^\star$ for the sequence ω.

ALGORITHM 1: Filling the minimum energy matrix

Entrée : ω – RNA of size n
Sortie : m – Matrix m, filled according to (1)

Fonction `FillMatrix` (ω)**:**
  $m \leftarrow$ `EmptyMatrix`$(n \times n)$
  // Initialize with 0 all the values of the diagonal up to θ.
  for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow i$ **to** $\min(i+\theta, n)$ **do**
      $m_{i,j} \leftarrow 0$
  for $i \leftarrow n$ **to** 1 **do**
    for $j \leftarrow i+\theta+1$ **to** n **do**
      ▶ Case A: Position i left without partner ;
      $m_{i,j} \leftarrow m_{i+1,j}$;
      ▶ Case B : Positions i and j form a base pair ;
      $m_{i,j} \leftarrow \min\left(m_{i,j}, m_{i+1,j-1} + E^{\omega}_{i,j}\right)$
      ▶ Case C: Position i paired to $k < j$;
      for $k \leftarrow i+\theta+1$ **to** $j-1$ **do**
        $m_{i,j} \leftarrow \min\left(m_{i,j}, m_{i+1,k-1} + m_{k+1,j} + E^{\omega}_{i,k}\right)$
  return m

ALGORITHM 2: Backtracking for the minimum energy structure.

Entrée : $[i, j]$ – Region under consideration
  m – Dyn. prog. matrix, previously computed according to Equation (1)
  ω – RNA of length n
Sortie : $S^{\star}$ – Structure minimizing free energy

Fonction `Backtrack` (i, j, m, ω)**:**
  if $j - i \leq \theta$ **then**
    return $\overbrace{\bullet \ldots \bullet}^{j-i+1}$ // The empty structure has minimum energy
  else
    ▶ Case A: Position i left without partner ;
    if $m_{i,j} = m_{i+1,j}$ **then** // Min. energy achieved by struct. where i is free
      $S^{\star}_{i} \leftarrow$ `Backtrack`$(i+1, j, m, \omega)$
      return $\bullet\, S^{\star}_{i}$
    ▶ Case B : Positions i and j form a base pair ;
    if $m_{i,j} = m_{i+1,j-1} + E^{\omega}_{i,j}$ **then** // Min. energy achieved by struct. pairing i and j
      $S^{\star}_{i,j} \leftarrow$ `Backtrack`$(i+1, j-1, m, \omega)$
      return $(\, S^{\star}_{i,j}\,)$
    ▶ Case C: Position i paired to $k < j$;
    for $k \leftarrow i+\theta+1$ **to** $j-1$ **do**
      if $m_{i,j} = m_{i+1,k-1} + m_{k+1,j} + E^{\omega}_{i,k}$ **then** // Optimal struct. pairing i and k
        $S^{\star}_{1} \leftarrow$ `Backtrack`$(i+1, k-1, m, \omega)$
        $S^{\star}_{2} \leftarrow$ `Backtrack`$(k+1, j, m, \omega)$
        return $(\, S^{\star}_{1}\,)\, S^{\star}_{2}$

2.1.1 Correctness of the algorithms

Proposition 1. *Algorithm 1 returns a matrix m which contains at position $m_{i,j}$ the minimum energy of the subsequence $\omega_{i,j}$.*

Proof. We shall show directly that, at each step of the computation, the value obtained for $m_{i,j}$ is correct. First, as mentioned in Section 1.2, there can be no base pair over a region of length smaller than $\theta + 2$. We can therefore initialize $m_{i,j}$ with 0 for all regions $[i, j]$ such that $j - i + 1 < \theta + 2$, equivalent to $j - i \leq \theta$, which is achieved by the double loop starting at line 3.

Now, in the loop starting at line 6, we fill in the matrix m one cell at a time. We iterate over the regions $[i, j]$ in ascending order on i (and then on j). In this way, we can guarantee that, by the time we compute an accurate $[i, j]$, the values $m_{i',j'}$ such that $i' < i$ have already been computed.

We are then going to assume these values $m_{i',j'}$, $i < i'$ are correct, and show that this implies the correction of $m_{i,j}$. In order to obtain the value of $m_{i,j}$, there are only three possible cases to consider:

Case A An optimal structure leaves position i without a partner. The optimal structure is then composed of the base pairs of an independent folding on $[i + 1, j]$, whose energy can be found in $\omega_{i+1,j}$. Since $i + 1 > i$, this value was already computed and can be assumed to be correct.

Case B An optimal structure pairs the positions i and j. In this case, the optimal structure is composed of a pair (i, j), of energy $E^{\omega}_{i,j}$, and an optimal folding over $[i + 1, j - 1]$, of energy found in $m_{i+1,j-1}$. It can be assumed that the latter is correctly computed because $i + 1 > i$.

Case C An optimal structure pairs the positions i and $k < j$. The pair (i, k) thus delimits two regions $[i + 1, k - 1]$ and $[k + 1, j]$, where the RNA forms independent folds (pseudoknots not being allowed). These two regions start after i, and their minimum energies can thus be found in $m_{i+1,k-1}$ and $m_{k+1,j}$. The sum of these terms is therefore completed by the contribution $E^{\omega}_{i,k}$ of the pair to obtain the minimum energy.

Given that any structure falls into one of these categories, the value assigned to $m_{i,j}$ indeed represents the structure having minimum energy over $[i, j]$. The correction of $m_{i,j}$ can thus be inferred and, by induction, the correction of the computation over any region ensues. □

Proposition 2. *Let m be the matrix computed by Algorithm 1, the function* `Backtrack`(i, j, m, ω) *of Algorithm 2 returns a minimal energy structure over the region $[i, j]$.*

Proof. By Proposition 1, we know that $m_{i,j}$ contains the minimum energy for any region $[i, j]$. Then there are four possible cases for the optimal structure:

Case 0: Sequence too short $i - j < \theta$. We know that if the regions contains less than $\theta + 2$ nucleotides, it cannot form a base pair. The structure without base pair, returned at line 2 is therefore the only compatible structure, and thus has minimal energy.

Case A: Position i left without a partner. In this case, we have $m_{i,j} = m_{i+1,j}$, and the optimal structure which starts with a free position, followed by an optimal structure over $[i + 1, j]$. Such a structure has energy $m_{i+1,j}$, and is thus also of minimal energy for $[i, j]$.

Case B: Paired positions i and j. In this case, we have $m_{i,j} = m_{i+1,j-1} + E^{\omega}_{i,j}$, with the latter coinciding with the energy of the structure pairing i to j, and forming an optimal folding on $[i + 1, j - 1]$, which the algorithm returns.

Case C: Position i paired to $k < j$. In this final case, we have that $m_{i,j} = m_{i+1,k-1} + m_{k+1,j} + E^{\omega}_{i,k}$. This optimal energy is indeed that of the structure, returned by the algorithm, containing the pair (i, k), and two optimal folds on regions $[i + 1, k - 1]$ and $[k + 1, j]$.

As the cases below cover all possible structures, we can conclude that the returned structure has indeed optimal energy over its region. □

2.1.2 Complexity analysis

The overall complexity of the above algorithm for producing a minimum energy secondary structure is $\Theta(n^3)$ in time, and $\Theta(n^2)$ in memory.

For `FillMatrix`, after allocating space for the matrix m in $\Theta(n^2)$, which bounds the memory complexity, the initialization takes $\Theta(n)$ time, due to the iterations of the innermost loop being bounded by a constant θ. The main contribution to the complexity is due to the three nested **for** loops (line 6 and following). The first two loops enumerate all regions $[i, j]$ such that $j - i > \theta$, and the last one chooses $k \in [i + \theta + 1, j - 1]$. Each of these loops is executed at most n times, and the time complexity is $\mathcal{O}(n^3)$, that is, asymptotically bounded by $C\, n^3$ where C is a constant.

To prove the asymptotic equivalent, and thus the complexity in $\Theta(n^3)$, one may consider triplets (i, k, j) associated with the innermost loop executions (line 10). One notices that such triplets correspond to choosing of 3 distinct elements among $n - C'$, C' being a constant, and are therefore counted by:

$$\binom{n - C'}{3} = \frac{(n - C')(n - C' - 1)(n - C' - 2)}{3!} = \frac{n^3}{6} + \mathcal{O}(n^2) \in \Theta(n^3).$$

To determine the complexity of `Backtrack`, it is clear that, excluding the recursive calls, the number of operations performed by the algorithm is linear on the size of the region $[i, j]$. Moreover, the recursive calls involve subregions whose cumulative size is decreasing. It follows that, in the tree of recursive calls, the total number of

iterations of the innermost loop, summed over all calls at depth p, remains bounded by n. Since the size of regions is strictly decreasing during successive recursive calls, the depth of the tree is bounded by n. The worst-case complexity is then $\Theta(n^2)$, and the complexity of `Backtrack` remains dominated by that of `FillMatrix`.

2.1.3 Going further

Despite its simplicity, this model already produces informative predictions, as can be seen in Section 4.2. They can also be substantially improved by considering a more realistic energy model (see Section 4.1). This requires a more complex algorithm, but very similar in principle to the one presented here.

Although the algorithm is commonly attributed to Nussinov et al. [1978], this seminal contribution was slightly different and, importantly, ambiguous: Despite being correct for the minimization, it did not allow the computation of the partition function introduced in Section 3.1. The version presented here is inspired by previous combinatorial works by Waterman [1978].

The algorithm can be also used to **predict the interaction of two RNAs**. Indeed, complexes can be predicted by running the algorithm on the concatenation of two RNAs, interspersed with θ anonymized nucleotides (`N`) to enable the full pairing of both RNAs. A minimum energy complex is then obtained, composed of both intramolecular base pairs (within a single RNA) and some intermolecular base pairs (involving both RNAs). However, since pseudoknots are forbidden, interactions remain limited to positions in the outer face of each of the two RNA (intramolecular) structures. More sophisticated dynamic programming schemes [Mückstein et al., 2006] have therefore have been introduced to capture more realistic conformation spaces, allowing for example the interaction of loops.

The algorithm can also be used to *simplify* a pseudoknotted structure $S^{\pitchfork}$, in order to recover a maximal non-crossing subset of base pairs. To this end, it is sufficient to adopt an energy model where $E^{\omega}_{i,j} := -1$ if $(i,j) \in S^{\pitchfork}$, and $E^{\omega}_{i,j} := +\infty$ otherwise. Minimizing the energy is then equivalent to maximizing the number of pairs retained from $S^{\pitchfork}$ such that pseudoknots are removed while arguably maximizing the residual structural information.

2.2 Listing (sub)optimal structures

Although fruitful, the energy minimization paradigm remains highly sensitive to the intrinsically-imperfect inaccuracy of energy models. It is potentially impacted by measurement errors involved in the energy parameters, including the individual contributions of base pairs and, more generally, structural motifs.

Consequently, a structure S may be marginally more stable than an alternative structure S', and still returned by an energy minimization algorithm. This may happen despite the energy distance $|E(S') - E(S)|$ being arbitrarily small, much smaller than the experimental imprecision of the protocols used to calibrate the energy model. In

such a case, it seems arbitrary to produce a structure S as representative of the folding process, especially when slightly suboptimal structures significantly differ.

This motivates the consideration of **Δ-admissible suboptimal structures**, *i.e.* structures compatible with the input RNA that located within at most Δ kcal.mol^{-1} of the minimum free energy structure. This problem was initially considered by Zuker [1989], in a version restricted to sets of structures having no pairwise base pairs in common. Nevertheless, due to its greedy nature, this strategy turned out to be highly dependent on the order of produced structures, and was found to overlook important stable structures.

A, more satisfactory, exhaustive version of the problem was subsequently considered by Wuchty et al. [1999].

Δ-suboptimal structures

Input: Sequence $\omega \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}^+$; Tolerance $\Delta \in \mathbb{R}^+$.
Output: Set $\mathcal{S}_\Delta$ of secondary Δ-admissible structures, having energy within Δ of minimum energy:

$$\mathcal{S}_\Delta = \{S \text{ such that } E(S) - \text{MFE}(\omega) \leq \Delta\}$$

A first idea, natural in the context of dynamic programming, consists in computing the exhaustive list of the Δ-sub optimal structures realizable over each region $[i, j]$. The lists associated with the different regions should then be stored in the cells of a specific matrix, and can be computed recursively. However, such a strategy would result in a memory complexity in $\Theta(n^3 \times M)$, where M is the number of Δ-sub optimal structures, and would quickly become prohibitive even for small RNAs.

Wuchty et al. [1999]'s algorithm modifies the backtrack phase to generate all suboptimal structures, while guaranteeing that each recursive call generates at least one admissible suboptimal structure. To this purpose, a parameter Δ, representing a **residual tolerance**, is introduced in the backtrack function. While inspecting cases in the dynamic programming, this parameter is used to decide whether or not a given case may contribute an admissible suboptimal. It is updated in the recursive calls to reflect the fact that choosing a given DP case may already *consume* some tolerance. The modified backtrack is summarized in Algorithm 3, and must be preceded by the DP computation of the MFE matrix (Algorithm 1) to obtain all the Δ suboptimal structures.

2.2.1 Correctness of the algorithm

Proposition 3. *For any tolerance $\Delta \geq 0$, and any region $\sigma := \{[1, n]\}$, Algorithm 3 returns the set of Δ-suboptimal structures such that $E(S) \leq m_{1,n} + \Delta$.*

Proof. Let us begin by noticing that, whenever invoked with $\Delta \geq 0$, `Subopts` will only pass positive values to Δ upon its subsequent recursive calls, as can be verified

ALGORITHM 3: Backtracking for Δ-suboptimal structures.

Entrée : σ – Set of regions being considered (initially $\sigma = \{[1, n]\}$)
S_p – Partial secondary structure (initially $S_p = \varnothing$)
Δ – Residual tolerance $\Delta \geq 0$
m – Dyn. prog. matrix, previously computed according to Equation (1)
ω – RNA of size n
Sortie : $\mathcal{S}_\Delta$ – Δ-sub optimal structures over the region $[i, j]$

Fonction `Subopts` $(\sigma, S_p, \Delta, m, \omega)$**:**
  if $\sigma = \varnothing$ **then**
    return $\{S_p\}$ // The partial structure S_p is Δ sub-optimal
  else
    $[i, j] \leftarrow$ `pop`(σ) // Removes the first region of the stack σ
    if $j - i \leq \theta$ **then**
      return `Subopts`$(\sigma, S_p, \Delta, m, \omega)$ // Processing other regions in σ
    else
      $\mathcal{A} \leftarrow \varnothing; \mathcal{B} \leftarrow \varnothing; \mathcal{C} \leftarrow \{\varnothing\}_{k=i}^{j}$
      ▶ Case A: Position i left without partner ;
      $\delta_i \leftarrow m_{i+1,j} - m_{i,j}$ // Minimum distance to optimal if i free
      if $\Delta - \delta_i \geq 0$ **then** // $\exists$ struct. Δ-sub optimal where i is free
        $\mathcal{A} \leftarrow$ `Subopts`$([i + 1, j] \circ \sigma, S_p, \Delta - \delta_i, m, \omega)$
      ▶ Case B : Positions i and j form a base pair ;
      $\delta_{i,j} \leftarrow (m_{i+1,j-1} + E^{\omega}_{i,j}) - m_{i,j}$ // Min. distance to opt. if (i, j) paired
      if $\Delta - \delta_{i,j} \geq 0$ **then** // $\exists$ struct. Δ-sub optimal pairing i and j
        $\mathcal{B} \leftarrow$ `Subopts`$([i + 1, j - 1] \circ \sigma, \{(i, j)\} \cup S_p, \Delta - \delta_{i,j}, m, \omega)$
      ▶ Case C: Position i paired to $k < j$;
      for $k \leftarrow i + \theta + 1$ **to** $j - 1$ **do**
        $\delta_{i,k} \leftarrow (m_{i+1,k-1} + m_{k+1,j} + E^{\omega}_{i,k}) - m_{i,j}$ // Min. dist. if (i, k) paired
        if $\Delta - \delta_{i,k} \geq 0$ **then** // $\exists$ struct. Δ-sub optimal pairing i and k
          $\sigma_k \leftarrow [i + 1, k - 1] \circ [k + 1, j] \circ \sigma$
          $\mathcal{C}_k \leftarrow$ `Subopts`$(\sigma_k, \{(i, k)\} \cup S_p, \Delta - \delta_{i,k}, m, \omega)$
      return $\mathcal{A} \cup \mathcal{B} \cup \bigcup_{k=i}^{j} \mathcal{C}_k$

in lines 11, 14, and 18. We thus assume without loss of generality that $\Delta \geq 0$, and consider the following generalization of Proposition 3.

Lemme 1. *Consider an RNA ω of length n, and opt the matrix of the minimal energies associated with regions. Then, for any list σ, any structure S_p, and any residual tolerance $\Delta \geq 0$, the call to* `Subopts`$(\sigma, S_p, \Delta, m, \omega)$ *returns the set of every structure S, compatible with ω and extending S_p with structures for every region of σ, such that*

$$E(S) \leq \Delta + \sum_{[i,j]\in\sigma} m_{i,j} + \sum_{(a,b)\in S_p} E^{\omega}_{a,b}. \tag{2}$$

To prove Lemma 1, let us consider the *cumulative length* $l(\sigma) := \sum_{[i,j]\in\delta} j-i+1$ of the regions in σ. First of all, it can be seen that, when $l(\sigma) = 0$ ($\sigma = \varnothing$) the structure S_p returned by the algorithm is such that

$$E(S_p) = \sum_{(a,b)\in S_p} E^{\omega}_{a,b} \leq \sum_{(a,b)\in S_p} E^{\omega}_{a,b} + \Delta$$

and thus satisfies the conditions of Equation (2). Moreover, it is the only possible extension of the input structure. This allows to conclude with the validity of Lemma 1 when $l = 0$, thus providing the base case of an inductive proof.

Next, we assume the correction of Lemma 1 for any list of regions σ having cumulative length $l < l^\star$, for any value $\Delta \geq 0$ and any structure S_p. Consider a list of regions $\sigma := [i, j] \circ \sigma'$ of cumulative length $l^\star$. The minimal energy accessible from a pair (σ, S_p) is

$$m(\sigma, S_p) := \sum_{[i,j]\in\sigma} m_{i,j} + \sum_{(a,b)\in S_p} E^{\omega}_{a,b}.$$

The choice of a decomposition case over $[i, j]$ can be seen as committing to a subset of structures, which may or may not include the local MFE, so the min. accessible energy is $m' \geq m(\sigma, S_p)$. In other words, an *optimality loss*

$$\delta := m' - m(\sigma, S_p) \geq 0$$

results from the choice of a decomposition case.

If $\delta > \Delta$ then, in any structure S resulting from successive recursive calls, one has $E(S) \geq m' = m(\sigma, S_p) + \delta > m(\sigma, S_p) + \Delta$, so S should not be returned by the algorithm. It follows that $\mathcal{A}$ (respectively $\mathcal{B}$ and $\mathcal{C}_k$) is empty when $\delta > \Delta$ (lines 11, 14 and 18).

When $\delta \leq \Delta$, the produced subset depends on the decomposition case:

Case A (i free): The minimal energy of a structure leaving i unpaired is given by

$$m([i{+}1, j] \circ \sigma', S_p) = \sum_{[x,y]\in\sigma'} m_{x,y} + \sum_{(a,b)\in S_p} E^{\omega}_{a,b} = m(\sigma, S_p) + m_{i+1,j} - m_{i,j}.$$

It thus follows that $\delta = m([i+1,j]\circ\sigma', S_p) - m(\sigma, S_p) = m_{i+1,j} - m_{i,j} \geq 0$. Since $l([i+1,j]\circ\sigma') = l(\sigma) - 1 < l^\star$, the induction hypothesis applies to the recursive call on σ' and S_p. This thus produces all the structures S extending S_p on $[i+1,j]\circ\sigma'$, such that

$$E(S) \leq m([i+1,j]\circ\sigma', S_p) + \Delta - \delta = m(\sigma, S_p) + \Delta.$$

The set $\mathcal{A}$ thus coincides with the restriction of the extensions of S_p over σ, where i is left unpaired.

Case B (i paired to j): The case where i is paired to j is similar, but induces a loss of optimality $\delta = E^\omega_{i,j} + m_{i+1,j-1} - m_{i,j}$. We still have $l([i+1,j-1]\circ\sigma') = l(\sigma) - 2 < l^\star$ and the induction hypothesis implies correcting the recursive call, which thus produces all the S structures, as extensions of $S_p \cup \{(i,j)\}$ on $[i+1,j-1]\circ\sigma'$ such that

$$E(S) \leq m([i+1,j-1]\circ\sigma', \{(i,j)\}\cup S_p) + \Delta - \delta = m(\sigma, S_p) + \Delta.$$

The structures in $\mathcal{B}$ thus extend S_p on σ and satisfy (2) while pairing i with j.

Case C (i paired to $k < j$) When i is paired to $k < j$, one has $\delta = E^\omega_{i,k} + m_{i+1,k-1} + m_{k+1,j} - m_{i,j}$. For any k, the recursive call is made on $\sigma_k := [i+1,k-1]\circ[k+1,j]\circ\sigma$, such that $l(\sigma_k) = l(\sigma) - 2$. The induction hypothesis thus applies, and the set of all structures extending $S_p \cup \{(i,k)\}$ over σ_k is obtained such that

$$E(S) \leq m(\sigma_k, \{(i,k)\}\cup S_p) + \Delta - \delta = m(\sigma, S_p) + \Delta.$$

The set $\mathcal{C}_k$ thus indeed represents the extensions of S_p on σ, verifying (2), and pairing i and k.

We remind that any structure over $[i,j]$ is generated by one of the three cases above. The assumed correctness of the algorithm, for any σ such that $l(\sigma) < l^\star$, thus extends to σ such that $l(\sigma) = l^\star$. In conjunction with the proven correctness when $l(\sigma) = 0$, this concludes the induction, and shows the validity of Lemma 1.

Finally, notice that for $\sigma = \{[1,n]\}$ and $S_p = \varnothing$ we obtain all structures such as

$$E(S) \leq \Delta + \sum_{[i,j]\in\sigma} m_{i,j} + \sum_{(a,b)\in S_p} E^\omega_{a,b} = m_{1,n} + \Delta$$

so the correction of Lemma 1 implies Property 3. □

2.2.2 Complexity analysis

The combinatorial explosion of the Δ-optimal structures produced by the algorithm, in exponential number on Δ and n, does not allow for a fine complexity analysis according to the input parameters only. The complexity is therefore considered as a

function of the number M of returned structures, and we show that `Subopts` can be executed in time $\mathcal{O}(M \times n^2)$.

We first focus on the structure of the tree T of recursive calls. Initially called with $\Delta \geq 0$, `Subopts` only makes recursive calls where $\Delta \geq 0$, as seen in lines 11, 14 and 18 of the pseudocode. Moreover, when $\sigma \neq \varnothing$ and $\Delta \geq 0$, at least one of the tests contribute a structure, so the leaves of T correspond to the case $\sigma = \varnothing$, producing a single structure $\{S_p\}$ (line 3). These structures are pairwise distinct (unambiguous decomposition), and thus in number M since any structure produced is propagated and backtracks to the root of T, where it is returned. Moreover, the height of T is at most n, since the cumulative size of the regions involved in σ is strictly decreasing during the successive recursive calls. The number of internal (non-root) nodes is therefore at most M because, at any depth $p \leq n$, the number of nodes at depth p is bounded by M (otherwise there would exist more than M leaves in T).

We obtain the predicted complexity of $\mathcal{O}(M \times n^2)$, noting that, excluding recursive calls, each run of `Subopts` requires, at worst, a linear number of elementary operations. For this purpose, suitable data structures will however have to be chosen, allowing addition to lists/stacks, and the set disjoint union in $\mathcal{O}(1)$ time. In practice, basic lists represent reasonable candidates, resulting in a relatively easy implementation.

2.2.3 Going further

The fundamental principle of the algorithm, which consists of updating a tolerance Δ according to the choices made during the backtrack, generalizes previous works by Waterman and Byers [1985], and can be slightly improved using techniques derived from natural language processing [Huang and Chiang, 2005].

The suboptimal backtrack can be adapted to any algorithm based on an unambiguous dynamic programming scheme. It remains valid for an ambiguous decomposition, albeit generating some structures redundantly. However, this multiplicity typically introduces an exponential overhead in n, thus restricting its practical use and motivating the search for alternative unambiguous DP schemes.

2.3 Comparative prediction: Simultaneous alignment/folding of RNAs

Comparative folding represents a final category of methods for structure prediction. It takes advantage of an evolutionary pressure towards structure conservation, observed within many functional families of non-coding RNAs. When a multiple sequence alignment is available for homologous RNAs, then a fruitful approach consists in folding the alignment, thereby simultaneously predicting a structure for all of its RNAs. This approach, which can be tackled using a variant of the Nussinov algorithm, optimizes the cumulative free energy while including substantial bonuses to reward **compensatory mutations**, *i.e.* pairs of positions in the alignment that mutate, yet preserve

the possibility to form base pairs.

Unfortunately, while the consideration of a structural alignment structure greater improves the quality of predictions, such an alignment may be difficult to build in the absence of a joint structure. This induces a circular dependence since the alignment depends on the structure, and vice versa, so it is unclear where to start (chicken and egg paradox). The pioneering work of [Sankoff, 1985], at the origin of multiple algorithms and dozens of methods and software, works around the issue by solving the folding and alignment problems simultaneously. More precisely, it introduces the problem of determining the alignment/structure pair that optimizes a combination of free-energy, conservation and compensatory mutations.

Similarly to multiple sequence alignment, the simultaneous alignment/folding problem is generally NP-hard [Wang and Jiang, 1994] for an arbitrary number of sequences, so a polynomial-time algorithm seems highly unlikely. Yet, the restriction of the problem to a pair of homologous RNAs is already relevant and informative. Indeed, an algorithm for the pairwise alignment can be leveraged in a a popular heuristics for the multiple RNA alignment problem, which progressively incorporates sequences into a partial multiple sequence alignment. Interestingly, the alignment/folding of an RNA pair admits an exact solution in $\Theta(n^6)$, based on a product of two dynamic programming schemes, visually described in Figure 5.

2.4 Joint alignment/folding model

Let us now describe more precisely the notion of **joint alignment/folding** of a pair (u, v) of RNAs. Let us recall that an **alignment** of two RNA sequences can be defined as a pair of character strings $A = (u', v')$ from an extended alphabet $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}, \mathtt{-}\}$, where the character - represents an insertion/deletion (indel) such that:

- The two sequences (u', v') have equal length $|u'| = |v'| \geq \max(|u|, |v|)$;
- u (resp. v) is obtained from u' (resp. v') by removing the indels (-).

For instance, the RNA sequences $u := \mathtt{ACGU}$ and $v := \mathtt{AGAU}$ admit (among others) the following alignments:

$A_1 :=$

	1	2	3		4
$u' \rightarrow$	A	C	G	-	U
$v' \rightarrow$	A	-	G	A	U
	1		2	3	4

$A_2 :=$

	1	2	3	4
	A	C	G	U
	A	G	A	U
	1	2	3	4

$A_3 :=$

	1	2	3	4				
	A	C	G	U	-	-	-	-
	-	-	-	-	A	G	A	U
					1	2	3	4

Each of the alignments induces a set of **correspondences**, called *(mis)matches*, each involving a positions in the two RNAs. For example, alignment A_1 above induces the set of matches $\{(1, 1), (3, 2), (4, 4)\}$, alignment A_2 induces the matches $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$ and A_3 has no matches ($\varnothing$).

Not all alignments are equally realistic, and evolutionary models can be inferred associate a probability with any alignment. Within the **maximal parsimony paradigm**, such probabilities are generally defined as product of independent probabilities, associated with evolutionary events suggested by the alignment.

For instance, the match $(1, 1)$ in A_1 suggests the presence of `A` in the (common) ancestral sequence, while the `C` in the second column could have been recently acquired by u (or lost by v). Finally, the matching of two distinct nucleotides, for example in the second column of A_2, suggests a mutation following the speciation/duplication of the RNA being considered.

The probabilities of these events can be estimated, and the probability of an alignment is obtained by multiplying the probabilities of the events implied by the columns of the alignment:

$$\mathbb{P}(A \mid u, v) = \prod_{\boxed{\begin{smallmatrix} x \\ y \end{smallmatrix}} \in A} \mathbf{P}\mu_{x,y} \prod_{\boxed{\begin{smallmatrix} x \\ \texttt{-} \end{smallmatrix}} \in A} \mathbf{P}\iota_x \prod_{\boxed{\begin{smallmatrix} \texttt{-} \\ y \end{smallmatrix}} \in A} \mathbf{P}\delta_y$$

where $\mathbf{P}\mu_{x,y}$ represents the probability of a conservation/match ($x = y$) or substitution/mismatch ($x \neq y$). Meanwhile, $\mathbf{P}\iota_x$ (resp. $\mathbf{P}\delta_y$) represents the probability of an insertion into u (resp. v).

A joint alignment/folding (A, S) then simply adds a secondary structure S on top of an alignment $A = (u', v')$, with each base pairs implicitly pairs nucleotides in the alignment columns. For any pair of bases (i, j) in S, at either (u'_i, u'_j) or (v'_i, v'_j) (or both) may form a base pair. In the case where both sequences can form the base pair, it possible to reward or penalize an apparent co-evolution of the two positions. On the contrary, if only one of the sequences allows pairing, then its presence in a shared structure becomes less likely.

To capture this aspect, we note $S \rightsquigarrow (U, V)$, where U and V are the restrictions of S_A, induced by u and v respectively, obtained by removing the indels (`-`) and the base pairs involving at least one indel. The probability for an alignment A, in conjunction with a common structure S for its two sequences, can then be (somewhat arbitrarily) defined as

$$\mathbb{P}(A, S \mid u, v) \propto \mathbb{P}(A \mid u, v)\, \mathbb{P}(U \mid u)\, \mathbb{P}(V \mid v) \prod_{(a,b) \in S} \mathbf{P}\pi^{u'_a, u'_b}_{v'_a, v'_b} \tag{3}$$

where $\mathbb{P}(S^\star \mid \omega)$ is the Boltzmann probability of $S^\star$ for a sequence ω, and $\mathbf{P}\pi^{x_1, y_1}_{x_2, y_2}$ is the probability of a **substitution of base pairs**, involving nucleotides (x_1, y_1) in u and (x_2, y_2) in v. This allows to reward *compensatory mutations*, defined here as:

Columns $\boxed{\begin{smallmatrix} x \\ y \end{smallmatrix}}$ and $\boxed{\begin{smallmatrix} a \\ b \end{smallmatrix}}$, $x \neq y, a \neq b$ such thats (x, a) and (y, b) can be paired.

Such mutations are often interpreted as indicating a positive selection pressure towards the formation of base pairs, and have been used in comparative RNA modeling since the early days of RNA research [Michel and Westhof, 1990].

This probability $\mathbb{P}(A, S \mid u, v)$ should be maximized, which is equivalent to maximizing the right-hand side of Equation (3). In practice, to avoid issues related to numerical precision, a logarithmic version of the objective function is considered. Since the logarithm is a monotonously increasing function, optimizing the logarithm of the objective function is equivalent to optimizing the objective function. Moreover, partition functions-induced terms contribute a constant factor that is independent of the structure or alignment. They can therefore be ignored for optimization purposes. The objective function then becomes

$$F(A,S) = \sum_{\boxed{\begin{smallmatrix} x \\ y \end{smallmatrix}} \in A} \mu_{x,y} + \sum_{\boxed{\begin{smallmatrix} x \\ - \end{smallmatrix}} \in A} \iota_x + \sum_{\boxed{\begin{smallmatrix} - \\ y \end{smallmatrix}} \in A} \delta_y - (E(U,u) + E(V,v)) + \sum_{(a,b) \in S} \pi^{u'_a, u'_b}_{v'_a, v'_b} \tag{4}$$

where μ, ι, δ and π represent the respective natural logarithms, multiplied by RT, respectively from $\mathbf{P}\mu$, $\mathbf{P}\iota$, $\mathbf{P}\delta$ and $\mathbf{P}\pi$.

Combined alignment/folding

Input: $u, v \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}^\star$; Matrices ι, δ, μ, and π.
Output: Alignment/structure pair $(A^\star, S^\star)$ having max probability w.r.t. (4) :

$$F(A^\star, S^\star) = \max_{A,S} F(A,S)$$

2.4.1 Algorithm and complexity

The above problem can also be solved using a polynomial dynamic programming algorithm. It relies on simulating all possible alignments during folding, as illustrated by Figure 5. A dynamic programming equation can be immediately derived to compute the maximal **logarithmic score** $f^{i,j}_{k,l}$ achievable on the region $[i, j]$ of u, and region $[k, l]$ of v.

Namely, for empty regions on u and/or v, we have:

$$f^{i,j<i}_{k,l<l} := 0 \qquad \text{// Regions of } u \text{ and } v \text{ both empty}$$

$$f^{i,j<i}_{k,l\geq k} := -m^v_{k,l} + \sum_{c \in v_{k,l}} \delta_c \qquad \text{// Empty region of } v \rightarrow \text{The region of } u \text{ is folded and inserted}$$

$$f^{i,j\geq i}_{k,l<k} := -m^u_{i,j} + \sum_{c \in u_{i,j}} \iota_c \qquad \text{// Empty region of } u \rightarrow \text{The region of } v \text{ is folded and deleted}$$

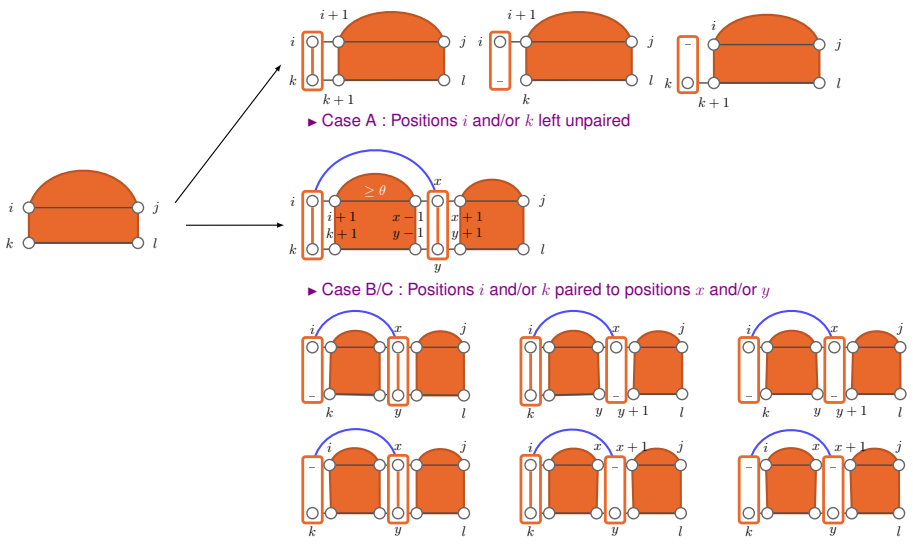


Figure 5: Decomposition of the combined alignment/folding space based on the Sankoff algorithm. For two RNA sequences restricted to regions $[i, j]$ and $[k, l]$, the decomposition distinguishes between leaving the first position unpaired, or pairing it to some other column.

where $m^{\omega}_{i,j}$ represents the minimum energy of a folding of ω over the $[i, j]$ region, computed as seen in Section 2.1. In addition to the energies induced by the independent foldings, the optimal scores must take into account the full insertion/deletion of the non-empty region, hence the above sums.

In the general case, non-empty regions are considered in both u and v, and one has

$$f^{i,j\geq i}_{k,l\geq k} := \max \begin{cases} \text{▸ Case A: Unpaired positions } i \text{ and/or } k \\ \max\limits_{b_i,b_k\in\{0,1\}^2} b_i\,\overline{b_k}\,\iota_{u_i} + \overline{b_i}\,b_k\,\delta_{v_k} + b_i\,b_k\,\mu_{u_i,v_k} + f^{i+b_i,j}_{k+b_k,l} \\ \text{▸ Case B/C: Pairings } (i,x) \text{ and/or } (k,y) \\ \max\limits_{\substack{\binom{x}{y}=\binom{i+\theta+1}{k+\theta+1};\\ \binom{b_i,b_x}{b_k,b_y}\in[0,1]^4 \\ b_ib_x+b_kb_y\geq 1}} \left| \begin{array}{l} b_i\,\overline{b_k}\,\iota_{u_i} + \overline{b_i}\,b_k\,\delta_{v_k} + b_x\,\overline{b_y}\,\iota_{u_x} + \overline{b_x}\,b_y\,\delta_{v_y} \\ + b_i\,b_k\,\mu_{u_i,v_k} + b_x\,b_y\,\mu_{u_x,v_y} \\ + b_i\,b_x\,b_k\,b_y\,\pi^{u_i,u_x}_{v_k,v_y} \\ + b_i\,b_x\,E^{u}_{i,x} + b_k\,b_y\,E^{v}_{k,y} \\ + f^{i+b_i,x-b_x}_{k+b_k,y-b_y} + f^{x+1,j}_{y+1,l} \end{array} \right. \end{cases} \tag{5}$$

where $\overline{b_p} := (1 - b_p)$ for any position p.

Each of the b_p reflects the implication ($b_p = 1$) or not ($b_p = 0$) of a position p in a column of the generated alignment. Such Boolean variables are used to factor the (otherwise numerous) cases in the decomposition, inferred by the enumeration of the

alignments, only incorporating relevant terms in each case.

For example, consider the term $b_i\, b_k\, \mu_{u_i,v_k}$. If the positions i and k are both aligned ($b_i = b_k = 1$), one then has $b_i\, b_k\, \mu_{u_i,v_k} = \mu_{u_i,v_k}$, corresponding to the conservation/substitution term expected for a match of i and k. On the other hand, if one of the two positions remains unaligned ($b_i = 0$ or $b_k = 0$), then one has $b_i\, b_k\, \mu_{u_i,v_k} = 0$, and the score does not include any contribution from the match.

Similarly, one has:

- $b_p\, \overline{b_q}\, \iota_{u_p}$ → Insertion score only if p without partner ($\overline{q}$);
- $\overline{b_p}\, b_q\, \delta_{u_q}$ → Deletion score only if q without partner ($\overline{p}$);
- $b_p\, b_q\, b_r\, b_s\, \pi_{v_r,v_s}^{u_p,u_q}$ → Base pair substitution score only if the four positions involved are pairwise aligned;
- $b_p\, b_q E_{p,q}^{\omega}$ → Energy of the base pair only if the pair is populated.

A variant of the Sankoff algorithm then computes the terms of this recurrence using dynamic programming. In its outermost loop, it processes (pairs of) regions by increasing order of their **cumulative size** $N(i,j,k,l)$, defined such that:

$$N(i,j,k,l) = |[i,j]| + |[k,l]| = j - i + k - l + 2$$

and in any order for the other indices/loops. A backtracking then completes the algorithm, and reconstructs the optimal folding/alignment.

Each run of this algorithm requires a $\Theta(n^6)$ time for two sequences u and v of equal length n, and $\Theta(n^4)$ in memory. More precisely, the time complexity of this algorithm is $\Theta(|u|^3 \cdot |v|^3)$, and its memory complexity is $\Theta(|u|^2 \cdot |v|^2)$. The decomposition underlying the Sankoff algorithm can also be generalized to M sequences, but the time complexity then increases to $\Theta(n^{3M})$, while the memory requirement scales to $\Theta(n^{2M})$.

2.4.2 Going further

The principle behind the Sankoff algorithm is at the core of virtually every approach for comparative prediction. The quality of its predictions is far superior to those obtained using energy minimization and (as of 2022) machine learning. However, its high complexity, especially in terms of memory, prevents its direct utilization for RNAs longer than $\sim$ 100 nucleotides. Consequently various computational tricks and heuristics can be found in modern implementations to support multiple (long) sequences without substantially sacrificing the predictive capability (see `SPARSE` [Will et al., 2015], currently at the state of the art despite its modest $\Theta(n^2)$ complexity).

The alignment model can be extended in a number of directions:

- First of all, the Sankoff algorithm can be extended to the full nearest-neighbor energy model introduced by [Turner and Mathews, 2010]. It can also support

more complex evolutionary models, for example taking a phylogenetic tree as input, in order to consider evolutionary distances and speciation events while interpreting a compensatory mutations.

- The cost associated with a sequence of g consecutive indels can be defined as an affine function $\alpha \times g + \beta$, instead of an implicit $\alpha' \times g$ in our model. The new objective function can be optimized using a variant of the Sankoff algorithm having the same complexity (up to constants) owing to a generic technique devised by Gotoh [1982]. A similar trick can be used to remove the ambiguity induced by the alignment (essentially due to the commutativity of the indels), unlocking the door to alignment under the assumption of preservation of the Boltzmann ensemble [Will et al., 2012].

3 Analyzing the Boltzmann ensemble

3.1 Computing the partition function

As seen in Section 2.2, suboptimal structures can be produced, and provide a sense of the diversity of almost-optimal structures. However, while they may be useful in modeling to suggest alternative structures, it is impossible to judge whether or not suboptimals accurately reflect the full diversity explored by the folding process. Indeed, at the thermodynamic equilibrium, the minimum-free energy (MFE) structure is typically associated with a probability which, despite being maximal by definition, remains abysmally small. Moreover suboptimal structures produced by the algorithm for a given tolerance Δ, may be extremely similar without being fully representative of the Boltzmann ensemble. It follows that, for a given tolerance, the list of suboptimals is typically biased towards the MFE and its trivial variations. It may overlook the existence of alternative conformations, represented by a large number of similar structures whose accumulated probability may exceed the MFE (+ variations) probability.

Let us formalize the notion of being representative of the folding space, also called **Boltzmann Ensemble**, using concepts from statistical mechanics. To this end, let us remind that at the thermodynamic equilibrium, the set of all possible structures $S \in \mathcal{S}_\omega$ for an RNA ω, is expected to follow a **Boltzmann distribution**:

$$\mathbb{P}(S \mid \omega) = \frac{e^{-\frac{E(S)}{RT}}}{\mathcal{Z}} \tag{6}$$

where T is the temperature (K), R is the Boltzmann constant (1.987×10^{-3} kcal.mol^{-1}.K^{-1}), and $\mathcal{Z}$ is the **partition function**, defined as

$$\mathcal{Z} = \sum_{S \in \mathcal{S}} e^{-E(S)/RT}. \tag{7}$$

The partition function is essential to adopt a statistical perspective over the Boltzmann ensemble. For example, the probability of the minimum free-energy structure,

giving us an idea of its prevalence, is given by $e^{-m_{1,n}/RT}/\mathcal{Z}$. More generally, computing the partition function is an essential prerequisite to sample the ensemble, as shown in Section 3.2, or to accurately compute average ensemble properties, as described in Section 3.3.

In practice, we must not only compute $\mathcal{Z}$, but also $\mathcal{Z}_{i,j}$ the partition function restricted to the set $\mathcal{S}_{i,j}$ of structures adopted on a region $[i,j]$. Note that $\mathcal{Z} := \mathcal{Z}_{1,n}$, so these (partial) partition functions can be used to find the global partition function of the system. Computation those for all regions $[i,j]$ therefore represents, a potentially complex, weighted counting problem defined as follows.

Computation of the partition function $\mathcal{Z}$

Input: Sequence $\omega \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{U}\}^+$, $|\omega| = n$.
Output: Partition function $\mathcal{Z}_{i,j}$ for any region $[i,j]$, defined as

$$\mathcal{Z}_{i,j} = \sum_{S \in \mathcal{S}_{i,j}} e^{-E(S)/RT}$$

While the number of terms in the sum grows exponentially with the sequence length n, it is in fact possible to compute $\mathcal{Z}$ very efficiently, in time only polynomial in n, as done by Algorithm 4.

In fact, we already introduced a correct polynomial algorithm for the problem in Section 2.1, up to a simple change of algebra! Indeed, our implementation of `FillMatrix` in Algorithm 1 can be slightly modified to calculate $\mathcal{Z}_{i,j}$ instead of $m_{i,j}$. Towards that, one simply has to replace sums by products, minimizations by sums, and transform constant energy terms into their Boltzmann factor:

$$\min \to + \qquad\qquad + \to \times \qquad\qquad E \to e^{-E/RT}$$

We finally obtain Algorithm 4, which computes the partition functions $\mathcal{Z}_{i,j}$.

3.1.1 Correctness of the algorithms

Let us start with a purely technical observation, which will represent the basis of our proof by induction. Precisely, let $E(S) = E_1 + \cdots + E_l$ be the energy of a structure S, which can be decomposed into l terms, one then has

$$\prod_{i=1}^{l} e^{-E_i/RT} = e^{-\sum_{i=1}^{l} E_i/RT} = e^{-E(S)/RT} \tag{8}$$

Equipped with this property, we can now establish the correction of the matrix filling procedure.

Proposition 4. *Algorithm 4 correctly computes the partition function of ω for each region.*

ALGORITHM 4: Computation of the partition function $\mathcal{Z}$

Entrée : ω – RNA of size n
Sortie : $\mathcal{Z}$ – Partition function Z

Fonction `PartitionFunction` (ω)**:**
  $\mathcal{Z} \leftarrow$ `EmptyMatrix`$(n \times n)$
  // Initialize to 1 all the values of the diagonal up to θ
  for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow i$ **to** $\min(i+\theta, n)$ **do**
      $\mathcal{Z}_{i,j} \leftarrow 1$
  for $i \leftarrow n$ **to** 1 **do**
    for $j \leftarrow i+\theta+1$ **to** n **do**
      ▶ Case A: Position i left without partner
      $\mathcal{Z}_{i,j} \leftarrow \mathcal{Z}_{i+1,j}$
      ▶ Case B: Positions i and j form a base pair
      $\mathcal{Z}_{i,j} \leftarrow \mathcal{Z}_{i,j} + \mathcal{Z}_{i+1,j-1} \times e^{-E^{\omega}_{i,j}/RT}$
      ▶ Case C: Position i paired to $k < j$
      for $k \leftarrow i+\theta+1$ **to** $j-1$ **do**
        $\mathcal{Z}_{i,j} \leftarrow \mathcal{Z}_{i,j} + \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \times e^{-E^{\omega}_{i,k}/RT}$
  return $\mathcal{Z}$

Proof. We proceed by *induction on the length of the region* $[i, j]$, and prove that the value computed in $\mathcal{Z}_{i,j}$ coincides with the partition function restricted to the subsequence $\omega_{i,j}$. In the *base case*, when $i - j \leq \theta$, there is only one possible structure, with no base pair and therefore zero energy, whose Boltzmann factor is $e^{-0/RT} = 1$. Now, the initialization assigns the value 1 to $\mathcal{Z}_{i,j}$ for any region of length at most $\theta + 1$, and we thus get the expected result.

We now assume the validity of the proposition for any region $[i', j']$ of length $j' - i' + 1 < n^\star$, meaning that the value $\mathcal{Z}_{i',j'}$ coincides well with the partition function restricted to $\omega_{i',j'}$. Now consider a region $[i, j]$ of length $n^\star$, while computing $\mathcal{Z}_{i,j}$ we have three possible cases:

Case A: Position i left without a partner. Since $j - i + 1 < n^\star$, the induction assumption applies and, in conjunction with the lack of energy contribution from the unmatched positions, implies that:

$$\mathcal{Z}_{i+1,j} = \sum_{S' \in \mathcal{S}_{i+1,j}} e^{\frac{-E(S')}{RT}} = \sum_{S' \in \mathcal{S}_{i+1,j}} e^{\frac{-E(\bullet S')}{RT}} = \sum_{S \in \bullet\mathcal{S}_{i+1,j}} e^{\frac{-E(S)}{RT}}.$$

In other words, $\mathcal{Z}_{i+1,j}$ coincides with the partition function on $[i, j]$, restricted to structures letting i free. We will denote this restriction by $\mathcal{Z}_{\bullet S}$ in the following.

Case B: Positions i paired with j. The induction hypothesis applies to $[i+1, j-1]$.

Noting that the pair (i,j) provides an energy $E^{\omega}_{i,j}$, we get:

$$e^{\frac{-E^{\omega}_{i,j}}{RT}} \times \mathcal{Z}_{i+1,j-1} = e^{\frac{-E^{\omega}_{i,j}}{RT}} \sum_{S' \in \mathcal{S}_{i+1,j-1}} e^{\frac{-E(S')}{RT}} = \sum_{S' \in \mathcal{S}_{i+1,j-1}} e^{\frac{-(E^{\omega}_{i,j}+E(S'))}{RT}}$$
$$= \sum_{S' \in \mathcal{S}_{i+1,j-1}} e^{\frac{-E((S'))}{RT}} = \sum_{S \in (S_{i+1,j-1})} e^{\frac{-E(S)}{RT}}$$

The computation proposed in the algorithm thus captures all the structures pairing i to j, whose partition function is denoted by $\mathcal{Z}_{(S)}$.

Case C: Position i paired to $k < j$. In this case, the algorithm adds to the partition function a quantity $\sum_{i+\theta+1}^{j-1} e^{-E^{\omega}_{i,j}/RT} \mathcal{Z}_{i+1,k-1}\mathcal{Z}_{k+1,j}$ with correct contributions as follows from the induction hypothesis. One thus obtains:

$$e^{\frac{-E^{\omega}_{i,k}}{RT}} \mathcal{Z}_{i+1,k-1}\mathcal{Z}_{k+1,j} = e^{\frac{-E^{\omega}_{i,k}}{RT}} \sum_{S_1 \in \mathcal{S}_{i+1,k-1}} e^{\frac{-E(S_1)}{RT}} \sum_{S_2 \in \mathcal{S}_{k+1,j}} e^{\frac{-E(S_2)}{RT}}$$
$$= \sum_{S_1 \in \mathcal{S}_{i+1,k-1}} \sum_{S_2 \in \mathcal{S}_{k+1,j}} e^{-\frac{E^{\omega}_{i,k}+E(S_1)+E(S_2)}{RT}} = \sum_{S \in (S_{i+1,k-1})\mathcal{S}_{k+1,j}} e^{\frac{-E(S)}{RT}}.$$

The term of the sum coincide with the definition of the partition function restricted to the structures paring i to k, for a given value of k. By summing over all the values of k on $[i+\theta+1, j-1]$, one obtains the partition function $\mathcal{Z}_{(S_1)S_2}$ of all the structures pairing i to any position other than j.

It is easy to see that the decomposition is unambiguous, *i.e.* that the various cases cover pairwise disjoint sets of structures. Moreover, any structure over a region $[i,j]$ falls into one of these three categories. We conclude that:

$$\mathcal{Z}_{\bullet S} + \mathcal{Z}_{(S)} + \mathcal{Z}_{(S_1)S_2} = \sum_{\substack{S \in \bullet \mathcal{S}_{i+1,j} \bigcup (\mathcal{S}_{i+1,j-1}) \\ \bigcup (S_{i+1,k-1})\mathcal{S}_{k+1,j}}} e^{-\frac{E(S)}{RT}} = \sum_{S \in \mathcal{S}_{i,j}} e^{\frac{-E(S)}{RT}}. \quad (9)$$

The validity of the computed partition function on any region of length $n < n^{\star}$ implies the correctness on regions of size $n^{\star}$, allowing us to conclude the induction. □

The differences between Algorithms 1 and 4 lead to constant time/memory overheads: elementary energy terms are $e^{-E/RT}$ instead of E while sums and products replace the minima and sums, all computable in constant time. We thus obtain an algorithm running in overall $\Theta(n^3)$ time and $\Theta(n^2)$ space complexity.

3.1.2 Going further

The change of algebra $(\min, +, E) \to (+, \times, e^{-E/RT})$ can, in principle, be adapted to any combinatorial problem solvable using dynamic programming. However, in

order for the modified algorithm to compute the true partition function, one must ensure that the underlying decomposition is complete – that it captures all the elements of the search space – and unambiguous – that it produces each element in a single way.

The number of secondary structures compatible with an RNA ω can also be easily obtained through a computation of a partition function. Indeed, assigning a very large value to the temperature T, and one obtains

$$\mathcal{Z} := \mathcal{Z}_{1,n} = \sum_{s \in \mathcal{S}} e^{-E(S)/RT} \underset{T \to +\infty}{\longrightarrow} \sum_{s \in \mathcal{S}} e^{0} = \sum_{s \in \mathcal{S}} 1 = |\mathcal{S}|.$$

3.2 Statistical sampling

The partition function is an essential quantity to derive the statistical properties of the folding space. However, it essentially only gives access to the individual probabilities of the structures. Meanwhile, there is a large number of structures, growing exponentially with the sequence size n, all associated with probabilities that are exponentially small. Computing the statistical properties in a deterministic fashion, by going through the list of all structures while accounting from their individual probabilities, would then require exponential time.

To overcome this issues, while still granting access to statistics of the Boltzmann ensemble for a specific RNA, Ding and Lawrence [2003] introduce a **random generation algorithm**, also called **stochastic sampling**. The idea is to modify the backtracking step in order to produce a random structure, generated according to the **Boltzmann distribution**

$$\mathbb{P}(S \mid \omega) = \frac{e^{-E(S)/RT}}{\mathcal{Z}}$$

where $\mathcal{Z} = e^{-E(S)/RT}$ is the partition function, calculated as shown in Section 3.1.

Sampling structures

Input: Sequence $\omega \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}^+$, $|\omega| = n$.
Output: Structure S with probability

$$\mathbb{P}(S) = \frac{p_S}{\mathcal{Z}} = \frac{e^{-E(S)/RT}}{\mathcal{Z}}$$

The **stochastic backtracking**, implemented as Algorithm 2, can be used to solve the problem in a simple energy model.

Its principle, illustrated by Figure 6, relies on a random choice, at each step of the generation, of a decomposition case with probability proportional to its contribution to the partition function. Namely, let us consider a region $[i, j]$ giving access to a set $\mathcal{S}_{i,j}$ of structures, and assume that each decomposition case gives access to a subset $\mathcal{S}' \subseteq \mathcal{S}_{i,j}$, associated with a partition function $\mathcal{Z}_{\mathcal{S}'} := \sum_{S \in \mathcal{S}'} e^{-E(S)/RT}$. Such a

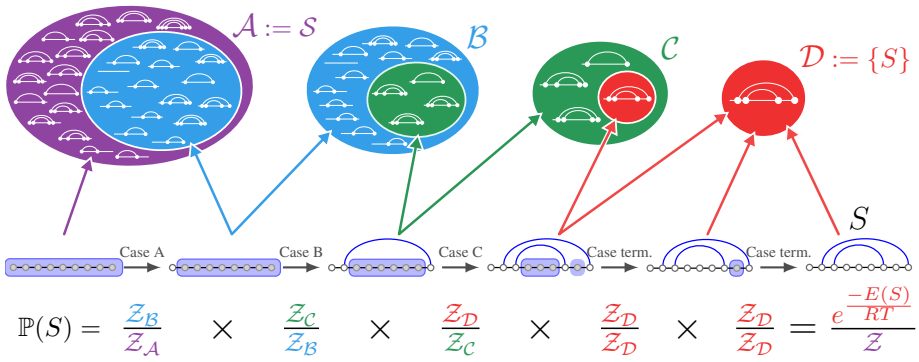


Figure 6: Principle of random generation using the recursive method. At each step, a decomposition case is chosen with probability proportional to its contribution in the partition function. The probability of generating a given structure S then equals the probability product over the chosen cases, and the consecutive numerators/denominators cancel out, leading S to the generated with Boltzmann probability.

case will be chosen, an backtracked upon, with probability:

$$p_{\mathcal{S}'} = \frac{\mathcal{Z}_{\mathcal{S}'}}{\mathcal{Z}_{i,j}}$$

If a valid Boltzmann generator for $\mathcal{S}'$ is available and called, the probability of emitting a given structure $S' \in \mathcal{S}'$ for $[i, j]$ then becomes

$$\mathbb{P}(S' \mid [i,j]) = p_{\mathcal{S}'} \times \mathbb{P}(S' \mid \mathcal{S}') = \frac{\mathcal{Z}_{\mathcal{S}'}}{\mathcal{Z}_{i,j}} \times \frac{e^{-E(S')/RT}}{\mathcal{Z}_{\mathcal{S}'}} = \frac{e^{-E(S')/RT}}{\mathcal{Z}_{i,j}}$$

where one recognizes the targeted probability when called for the full sequence ω with $[i, j] = [1, n]$.

3.2.1 Correctness of the algorithm

Proposition 5. *Let $\mathcal{Z}_{i,j}$ be the partition function of an RNA ω restricted to $[i, j]$. Then a call to* `RandomStruct`$(i, j, \omega, \mathcal{Z})$ *returns some $S^\star \in \mathcal{S}_{i,j}$ with probability:*

$$\mathbb{P}(S^\star) = \frac{e^{-E(S^\star)/RT}}{\mathcal{Z}_{i,j}}$$

Proof. We proceed by *induction on the length of the regions* $[i, j]$. In the *base case*, when $i - j \leq \theta$, there is only one possible structure $S^\star$, the empty structure. It is returned with probability 1 as described at line 2.

ALGORITHM 5: Generates a structure S with probability $e^{-E(S)/RT}/\mathcal{Z}$

Entrée : $[i, j]$ – Region being considered
$\mathcal{Z}$ – Partition function for each region, computed by Algorithm 4
ω – RNA of size n
Sortie : S – Random Boltzmann-distributed structure compatible with ω over $[i, j]$

Fonction `RandomStruct`$(i, j, \mathcal{Z}, \omega)$**:**
  if $j - i \leq \theta$ **then return** $\overbrace{\bullet \ldots \bullet}^{j-i+1}$ // Empty structure is unique → Probability 1
  else
    $a \leftarrow$random$(0, \mathcal{Z}_{i,j})$ // Random number, uniform drawn in $[0, \mathcal{Z}_{i,j}[$
    ▶ Case A: Position i left without partner
    $a \leftarrow a - \mathcal{Z}_{i+1,j}$ // Subtracting part. func. of all structures leaving i unpaired
    if $a < 0$ **then** // True when $a \in [0, \mathcal{Z}_{i+1,j}[$ → Probability $\mathcal{Z}_{i+1,j}/\mathcal{Z}_{i,j}$
      $S_i^\star \leftarrow$ `RandomStruct`$(i + 1, j, \mathcal{Z}, \omega)$
      return $\bullet\, S_i^\star$
    ▶ Case B: Positions i and j form a base pair
    $a \leftarrow a - \mathcal{Z}_{i+1,j-1} \times e^{-E^\omega_{i,j}/RT}$ // Subtracting part. func. of all structures pairing i to j
    if $a < 0$ **then** // True with prob. $\mathcal{Z}_{i+1,j} \times e^{-E^\omega_{i,j}/RT}/\mathcal{Z}_{i,j}$
      $S^\star_{i,j} \leftarrow$ `RandomStruct`$(i + 1, j - 1, \mathcal{Z}, \omega)$
      return ($S^\star_{i,j}$ **)**
    ▶ Case C: Position i paired to $k < j$;
    for $k \leftarrow i + \theta + 1$ **to** $j - 1$ **do**
      $a \leftarrow a - \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \times e^{-E^\omega_{i,k}/RT}$ // Sub. part. fun. pairing i to $k < j$
      if $a < 0$ **then** // True with prob. $(\mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \times e^{-E^\omega_{i,k}/RT})/\mathcal{Z}_{i,j}$
        $S_1^\star \leftarrow$ `RandomStruct`$(i + 1, k - 1, \mathcal{Z}, \omega)$
        $S_2^\star \leftarrow$ `RandomStruct`$(k + 1, j, \mathcal{Z}, \omega)$
        return ($S_1^\star$ **)** $S_2^\star$

Let us now assume that the property holds for any region $[i,j]$ such that $j-i+1 \leq n-1$. As we have seen for the computation of the partition function, we have:

$$\overbrace{\mathcal{Z}_{i,j}}^{S} = \overbrace{\mathcal{Z}_{i+1,j}}^{\bullet\mathcal{S}_{i+1,j}} + \overbrace{E^{\omega}_{i,j} \times \mathcal{Z}_{i+1,j-1}}^{(\mathcal{S}_{i+1,j-1})} + \sum_{k=i+\theta+1}^{j-1} \overbrace{E^{\omega}_{i,k} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j}}^{(\mathcal{S}_{i+1,k-1})\mathcal{S}_{k+1,j}}. \quad (10)$$

Each term of the sum represents the contribution of a subset of possible structures for $S^\star$, associated with one of the decomposition cases. The random number a generated by Algorithm 5 is be used to identify the decomposition case, so that each ends up being chosen with a probability proportional to its contribution.

Consider a structure $S^\star$ generated by the algorithm over a region $j-i+1 \leq n$. There are three possible cases:

Case A: Position i without partner in $S^\star$. One has $S^\star = \bullet S'$, where S' is generated over $[i+1,j]$ with Boltzmann probability as per the induction hypothesis, and such that $E(S^\star) = E(S')$. Moreover, the probability of choosing this case is $\mathbb{P}(a \leq \mathcal{Z}_{i+1,j}) = \mathcal{Z}_{i+1,j}/\mathcal{Z}_{i,j}$. The emission probability of generating $S^\star$ is therefore:

$$\mathbb{P}(S^\star) = \frac{\mathcal{Z}_{i+1,j}}{\mathcal{Z}_{i,j}} \cdot \mathbb{P}(S') = \frac{\mathcal{Z}_{i+1,j}}{\mathcal{Z}_{i,j}} \cdot \frac{e^{\frac{-E(S')}{RT}}}{\mathcal{Z}_{i+1,j}} = \frac{e^{\frac{-E(S')}{RT}}}{\mathcal{Z}_{i,j}} = \frac{e^{\frac{-E(S^\star)}{RT}}}{\mathcal{Z}_{i,j}}.$$

Case B: Positions i and j paired in $S^\star$. One has $S^\star = (S')$, where S' is generated over $[i+1,j-1]$ such that $E(S^\star) = E(S') + E^{\omega}_{i,k}$. The probability of choosing a such that it identifies this case is $\mathcal{Z}_{i+1,j-1} \times e^{-E^{\omega}_{i,j}/RT}/\mathcal{Z}_{i,j}$. The likelihood of generating $S^\star$ is therefore:

$$\mathbb{P}(S^\star) = \frac{\mathcal{Z}_{i+1,j-1} \times e^{\frac{-E^{\omega}_{i,j}}{RT}}}{\mathcal{Z}_{i,j}} \times \frac{e^{\frac{-E(S')}{RT}}}{\mathcal{Z}_{i+1,j-1}} = \frac{e^{-\frac{E(S')+E^{\omega}_{i,j}}{RT}}}{\mathcal{Z}_{i,j}} = \frac{e^{\frac{-E(S^\star)}{RT}}}{\mathcal{Z}_{i,j}}.$$

Case C: Position i paired to $k<j$ in $S^\star$. In the last case, we have $S^\star = (S_1)S_2$, where S_1 and S_2 are generated over regions $[i+1,k-1]$ and $[k+1,j]$ respectively, and $E(S^\star) = E^{\omega}_{i,k} + E(S_1) + E(S_2)$. The probability of choosing this case is then

$$\frac{\mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \times e^{-E^{\omega}_{i,k}/RT}}{\mathcal{Z}_{i,j}}$$

and it follows that the probability of generating $S^\star$ is

$$\begin{aligned}\mathbb{P}(S^\star) &= \frac{\mathcal{Z}_{i+1,k-1}\,\mathcal{Z}_{k+1,j}\,e^{\frac{-E^{\omega}_{i,k}}{RT}}}{\mathcal{Z}_{i,j}} \frac{e^{\frac{-E(S_1)}{RT}}}{\mathcal{Z}_{i+1,k-1}} \frac{e^{\frac{-E(S_2)}{RT}}}{\mathcal{Z}_{k+1,j}} \\ &= \frac{e^{-\frac{E^{\omega}_{i,k}+E(S_1)+E(S_2)}{RT}}}{\mathcal{Z}_{i,j}} = \frac{e^{-E(S^\star)/RT}}{\mathcal{Z}_{i,j}}.\end{aligned}$$

As these three cases cover exhaustively and uniquely all the structures, the function `RandomStruct`$(i, j, \mathcal{Z}, \omega,)$ thus returns $S^\star \in \mathcal{S}_{i,j}$ with the expected probability. $\square$

3.2.2 Complexity

Assuming that a random uniform number can be generated in constant time, the complexity of the `RandomStruct` algorithm is $\Theta(n^2)$, with a worst case similar to the one analyzed in Section 2.1. It is thus possible to generate a sample of M structures in $\Theta(M.n^2)$ time after a preprocessing in $\Theta(n^3)$ time and $\Theta(n^2)$ space.

3.2.3 Going further

From a representative sample of structures, it is possible to estimate the statistical properties of a folding. For instance, to study how structured an RNA is, one can use the expected number of base pairs as a proxy, estimated from a random sample of structures $S_1, S_2, ..., S_M$ through a basic estimator:

$$\mathbb{E}(\#\text{Pairs}(S)) = \frac{\sum_{i=1}^{M} \#\text{Pairs}(S_i)}{M}$$

Sampling can also be combined with unsupervised machine learning (*clustering* algorithm), based on a notion of base pair distance, to identify dominant conformation(s) within the Boltzmann Ensemble [Ding et al., 2005].

The average complexity of the algorithm is $\Theta(n\sqrt{n})$ [Ponty, 2008]. It can be significantly improved by simply changing the examination order of the k values in case C. For this, the original loop order

$$k := i + \theta + 1 \to i + \theta + 2 \rightsquigarrow \ldots \to j - 2 \to j - 1$$

can simply be replaced by a Boustrophedon order, converging from the extremities of the interval towards its center

$$k := i + \theta + 1 \to j - 1 \to i + \theta + 2 \to j - 2 \to \ldots$$

A highly technical analysis of the worst-case complexity allows to conclude that the worst-case execution time then becomes $\mathcal{O}(n \log n)$ [Ponty, 2008].

3.3 Boltzmann probability of structural patterns

Statistical sampling enables the estimation of statistical properties at thermodynamic equilibrium, while offering (probabilistic) guarantees regarding their accuracy, for instance in the form of confidence intervals. It thus makes it possible for example, by generating enough structures, to satisfactorily address questions such as: *What is the average energy of a folding at thermodynamic equilibrium?*

However, confidence intervals, based on the law of large numbers, only allow to control the absolute error. Sampling encounters issues, or becomes very costly,

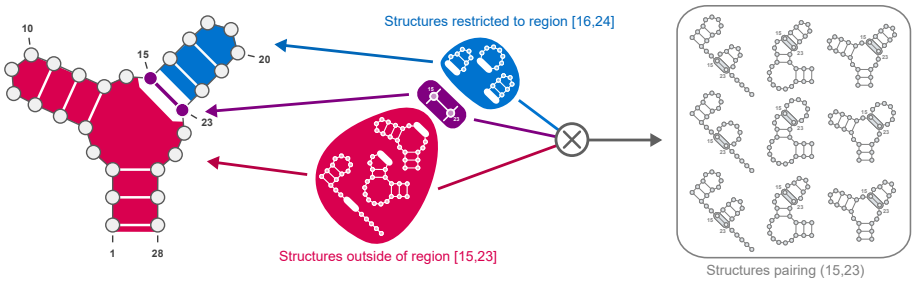


Figure 7: Decomposition of structures featuring a structural pattern, here the base pair $(15, 23)$. Any structure featuring $(15, 23)$ contains an outside part (red), and an inside part (blue), both of which independently contribute to the stability. The set of structures containing the pair can be obtained as a Cartesian product of both, considering all combinations of inner and outer structures.

when the objective is to estimate quantities having smaller, or very diverse values. In particular, it does not provide a very satisfactory solution to the question: *What are the Boltzmann probabilities of all base pairs* (i, j)*?* More generally, it provides only, possibly noisy, probabilistic estimates for computing the probability of complex structural patterns at the thermodynamic equilibrium.

A major contribution of the seminal paper by McCaskill [1990] resides in an efficient computation of the **exact Boltzmann probability of a structural pattern** m:

$$\mathbb{P}(m \in S) := \sum_{\substack{S \in \mathcal{S} \\ \text{such that } m \in S}} \mathbb{P}(S \mid w) = \sum_{\substack{S \in \mathcal{S} \\ \text{such that } m \in S}} \frac{e^{-E(S)/RT}}{\mathcal{Z}} = \frac{\mathcal{Z}_m}{\mathcal{Z}}$$

where $\mathcal{Z}$ is the partition function and $\mathcal{Z}_m := \sum_{S \in \mathcal{S}; m \in S} e^{-E(S)/RT}$ is the partition function restricted to structures featuring the motif m. Since $\mathcal{Z}$ is computable in $\Theta(n^3)$ time, as seen in Section 3.1, the main remaining difficulty resides in computing $\mathcal{Z}_m$.

The algorithm proposed by McCaskill adapts the approach of the **Inside-Outside** algorithm [Lari and Young, 1990], initially proposed in the context of automatic language processing. It is based on a non-ambiguous decomposition, illustrated in Figure 7, of all S structures containing $m \in S$ into:

- A decomposition-induced **production** P, applicable to a region $[i, j]$, creating an instance of the pattern m, and followed by one or more substructure(s) over region(s) $[i_1, j_1], [i_2, j_2] \ldots$;
- An **(inside) structure** S_r for each region $[i_r, j_r]$ produced by P;
- An **outside structure** S_E, defined over $[1, n]$ while leaving a *hole* in $[i, j]$.

Proposition 6. *Let $E(P)$ be the proper contribution of a production P to the free energy, and $\mathcal{Y}_{i,j}$ the* ***outside partition function*** *with respect to the region $[i,j]$, such that*

$$\mathcal{Y}_{i,j} := \sum_{S_E \text{ external to } [i,j]} e^{\frac{-E(S_E)}{RT}} \tag{11}$$

One then has

$$\mathcal{Z}_m = \sum_{\substack{P=([i,j]\to[i_1,j_1],\dots,[i_r,j_r]) \\ \text{such that } m\in P}} e^{\frac{-E(P)}{RT}} \mathcal{Y}_{i,j} \prod_r \mathcal{Z}_{i_r,j_r}. \tag{12}$$

Proof. Let us first note that, for any structure S featuring m, one has $E(S) = E(P) + E(S_E) + \sum_r E(S_r)$. Consider then the quantity

$$\Phi := \sum_{\substack{P=[i,j]\to[i_1,j_1],[i_2,j_2]\dots \\ \text{such that } m\in P}} e^{\frac{-E(P)}{RT}} \mathcal{Y}_{i,j} \prod_r \mathcal{Z}_{i_r,j_r}.$$

By replacing the partition functions by their respective definitions, one obtains

$$\begin{aligned}\Phi &= \sum_{P;m\in P} e^{\frac{-E(P)}{RT}} \left(\sum_{\substack{S_E \text{ ext.} \\ \text{to } [i,j]}} e^{\frac{-E(S_E)}{RT}} \right) \prod_r \left(\sum_{\substack{S_r \\ \text{over } [i_r,j_r]}} e^{\frac{-E(S_r)}{RT}} \right) \\ &= \sum_{P;m\in P} \sum_{S_E,S_1,S_2,\dots} e^{-\frac{E(P)+E(S_E)+\sum_r E(S_r)}{RT}} = \sum_{\substack{S\in\mathcal{S} \\ \text{such that } m\in S}} e^{-E(S)/RT} \equiv \mathcal{Z}_m.\end{aligned}$$

□

Remind that the (inside) partition functions $\mathcal{Z}_{i,j}$ can be computed in time $\Theta(n^3)$ (see Section 3.1), simultaneously for all regions $[i,j]$. The only missing ingredient to compute $\mathcal{Z}_m$, and thus p_m, is the outside partition function $\mathcal{Y}$.

Outside partition function

Input: Sequence $\omega \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{U}\}^+$
Output: The **outside partition function** $\mathcal{Y}$ associated with each region $[i,j]$, defined as

$$\mathcal{Y}_{i,j} = \sum_{S_E \text{ outside } [i,j]} e^{-E(S_E)/RT}$$

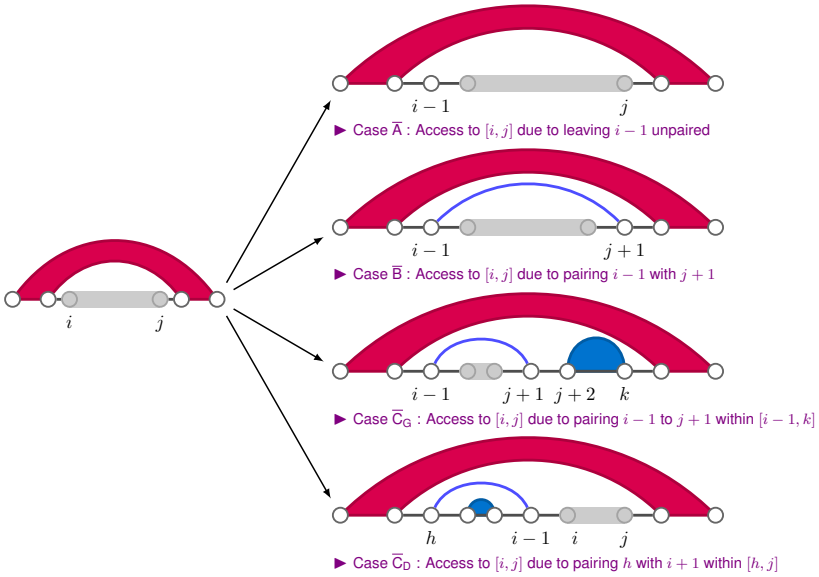


Figure 8: Decomposition of structures outside a region $[i, j]$.

Fortunately, $\mathcal{Y}$ follows a relatively simple formula, based on the decomposition described in Figure 8, which we establish by **inverting the rules** of the dynamic programming scheme. One then obtains, for $i > 1$:

$$\mathcal{Y}_{i,j} = \sum \begin{cases} \mathcal{Y}_{i-1,j} & \blacktriangleright \text{ Case } \bar{\mathsf{A}}\text{: Pos. } i-1 \text{ is free} \\ \text{// If } j < n \text{ and } j - i > \theta : & \blacktriangleright \text{ Case } \bar{\mathsf{B}}\text{: Pair } (i-1, j+1) \\ \quad e^{\frac{-E^{\omega}_{i-1,j+1}}{RT}}\, \mathcal{Y}_{i-1,j-1} & \\ \text{// If } j - i > \theta\text{:} & \blacktriangleright \text{ Case } \bar{\mathsf{C}}_{\mathsf{G}}\text{: Pair } (i-1, j+1) \\ \quad \sum_{k=j+2}^{n} e^{\frac{-E^{\omega}_{i-1,j+1}}{RT}}\, \mathcal{Y}_{i-1,k}\, \mathcal{Z}_{k+1,j} & \qquad \text{in } [i-1, k > j] \\ \sum_{h=1}^{i-\theta-2} e^{\frac{-E^{\omega}_{h,i-1}}{RT}}\, \mathcal{Y}_{h,j}\, \mathcal{Z}_{h+1,i-2} & \blacktriangleright \text{ Case } \bar{\mathsf{C}}_{\mathsf{D}}\text{: Pair } (h, i-1) \\ & \qquad \text{in } [h < i, j] \end{cases} \tag{13}$$

with $\mathcal{Y}_{1,n} := 1$ and $\mathcal{Y}_{1,j<n} := 0$. This equation can be computed for any interval using dynamic programming as described in Algorithm 6. The resulting algorithm has a time complexity in $\Theta(n^3)$, and a memory complexity in $\Theta(n^2)$.

ALGORITHM 6: Outside partition function

Entrée : ω – RNA of size n
Sortie : $\mathcal{Z}$ – Matrix $\mathcal{Y}$, filled according to Equation (13)

$\mathcal{Y} \leftarrow$ `EmptyMatrix`$(n \times n)$
// Initialize to 0 all the values of the diagonal up to θ
for $j \leftarrow 1$ **to** $n-1$ **do** $\mathcal{Y}_{1,j} \leftarrow 0$
$\mathcal{Y}_{1,n} \leftarrow 1$
for $i \leftarrow 2$ **to** n **do**
  for $j \leftarrow i$ **to** n **do**
    ▶ Case $\overline{\text{A}}$: Position i left without partner
    $\mathcal{Y}_{i,j} \leftarrow \mathcal{Y}_{i-1,j}$
    ▶ Case $\overline{\text{B}}$: Positions i and j form a base pair
    if $j < n\ j - i > \theta + 2$ **then**
      $\mathcal{Y}_{i,j} \leftarrow \mathcal{Y}_{i,j} + e^{-E^{\omega}_{i-1,j+1}/RT} \times \mathcal{Y}_{i-1,j-1}$
    ▶ Case $\overline{\text{C}_{\text{G}}}$: Position i paired to $k < j$
    if $j < n\ j - i > \theta + 2$ **then**
      for $k \leftarrow j+2$ **to** n **do**
        $\mathcal{Y}_{i,j} \leftarrow \mathcal{Y}_{i,j} + \mathcal{Y}_{i-1,k} \times e^{-E^{\omega}_{i-1,j+1}/RT} \times \mathcal{Z}_{k+1,j}$
    ▶ Case $\overline{\text{C}_{\text{D}}}$: Position i paired to $k < j$
    for $h \leftarrow 1$ **to** $i - \theta - 2$ **do**
      $\mathcal{Y}_{i,j} \leftarrow \mathcal{Y}_{i,j} + \mathcal{Y}_{h,j} \times e^{-E^{\omega}_{h,i-1}/RT} \times \mathcal{Z}_{h+1,i-2}$
return $\mathcal{Y}$

To calculate the Boltzmann probability of a pattern, the transitions producing a particular pattern thus remain to be enumerated. In practical terms, the probability of leaving a position i free is given by

$$\mathbb{P}(i \text{ free}) = \frac{\mathcal{Y}_{i,i} + \sum_{j=i+1}^{n} \mathcal{Y}_{i,j}\mathcal{Z}_{i+1,j}}{\mathcal{Z}_{1,n}}.$$

Similarly, the probability of forming a base pair (i, j) is obtained by

$$\mathbb{P}(\text{pair } (i,j)) = \frac{e^{\frac{-E^{\omega}_{i,j}}{RT}}\mathcal{Y}_{i,j}\mathcal{Z}_{i+1,j-1} + \sum_{k=j+1}^{n} e^{\frac{-E^{\omega}_{i,j}}{RT}}\mathcal{Y}_{i,k}\mathcal{Z}_{i+1,j-1}\mathcal{Z}_{j+1,k}}{\mathcal{Z}_{1,n}}.$$

Here, these probabilities can typically be computed simultaneously for all possible positions of the pattern in $\mathcal{O}(n^3)$ time.

3.3.1 Going further

In order to take advantage of an efficient algorithm, here in $\mathcal{O}(n^3)$ time, the pattern must be identifiable in the dynamic programming scheme. In the Nussinov decomposition, this constraint limits the list of eligible patterns to base pairs and unpaired

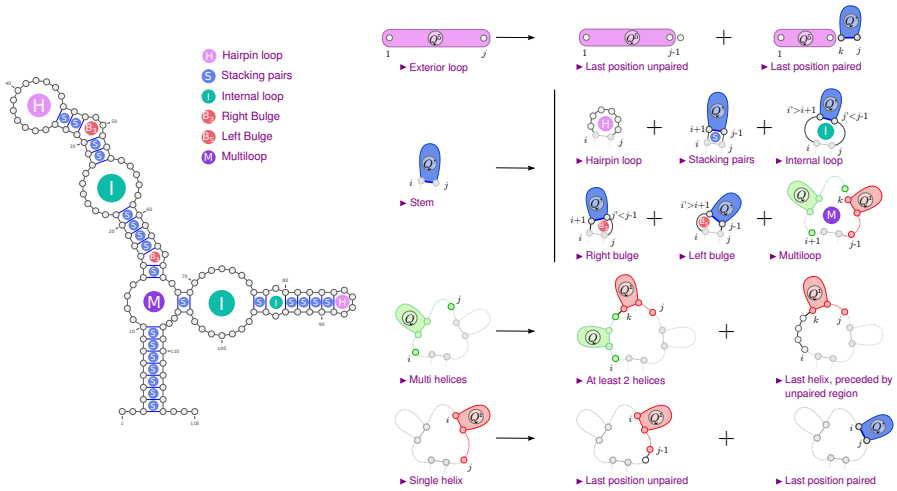


Figure 9: Loops of the Turner energy model, illustrated using the structure of a 5s ribosomal RNA (left), and decomposition (right) of all secondary structures allowing for the expression of the loops-based energy model.

positions. However, different types of loops can also be considered by more complex decompositions, for instance those capturing the Turner energy model as seen in Section 4.1.

The framework described above considers patterns (base pairs, unpaired positions) that appear at most once in each structure. However, the same calculation can be exactly used for a pattern m' that possibly occurs several times in a given structure. The value returned by the algorithm then simply becomes the **expectation of the number of occurrences of** m' in the Boltzmann distribution.

4 Studying RNA structure in practice

4.1 The Turner model

The base-pair based free energy model used throughout this chapter may appear over simplistic, and indeed is! Base pairs are actually not the main determinants of free energy, but the latter is thought to be dominated by the contributions of structural "blocks" [Xia et al., 1998] called **loops**, *namely* the closed regions appearing in the graph drawing of the secondary structure (see Figure 9). In particular, the **stackings** of two directly nested base pairs $(i, j) \rightarrow (i + 1, j - 1)$, often represent the main contributors to the free energy. Energies associated with the different types and contents of the loops, have been precisely calculated and extrapolated from the results of optical melting curve experiments [Turner and Mathews, 2010].

Although more complex, this energy model preserves a notion of **independence** of local patterns in the structure. For this reason, it can be captured by a decomposition, illustrated in Figure 9, which retains the same properties (completeness, unambiguity, correctness) as the simple base pairs-based decompositions presented in this chapter. All the methods and algorithmic approaches presented in this chapter can thus be adapted, with essentially the same complexity, to this more realistic model.

The resulting gain in predictive accuracy is significant, as illustrated in Figure 10. We considered a structure/sequence database proposed by Mathews [2004], comprising RNAs having known structure. For each sequence, we used Algorithm 2 to produce a minimum energy structure, based on contributions `G-C` $\rightarrow$ -3, `A-U` $\rightarrow$ -2, and `G-U` $\rightarrow$ -1 (kcal.mol^{-1}). We also run a recent version of the `RNAfold` software [Lorenz et al., 2011], implementing energy minimization in the Turner model, to produce an alternative, hopefully more accurate, structure.

To evaluate the quality of a predicted structure S, for an RNA ω of known structure $S^\star$, we considered the True/False Positive/Negative (see Section **??** of Chapter **??**) base pairs, such that:

$$\text{VP} := |S \cap S^\star| \quad \text{FP} := |S \setminus S^\star| \quad \text{VN} := |\{(i,j)\} \setminus (S \cup S^\star)| \quad \text{FN} := |S^\star \setminus S|.$$

The **sensitivity** is then derived, defined as the proportion of pairs from the reference $S^\star$ that are actually predicted by a given algorithm; conversely, the **positive predictive value** (PPV) is the proportion of pairs from S, predicted by the algorithm, that are found in the reference; and finally the **Matthews Correlation Coefficient** (MCC), an agglomeration of the different measures, such that:

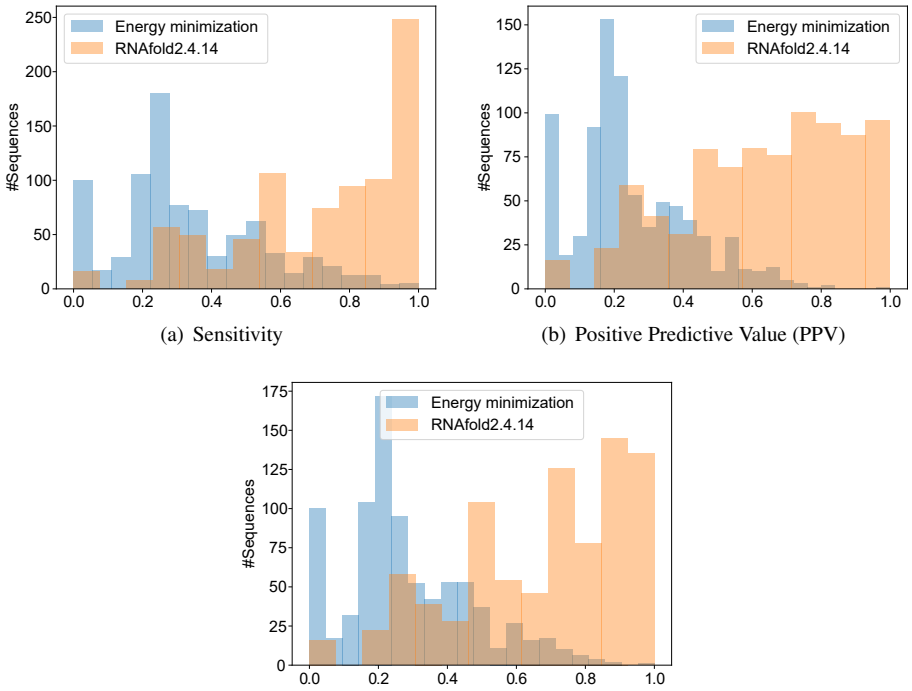
$$\text{Sens} = \frac{\text{VP}}{\text{VP} + \text{FN}} \qquad \text{PPV} = \frac{\text{VP}}{\text{VP} + \text{FP}}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \approx \sqrt{\text{Sens.} \times \text{PPV}}.$$

With respect to these metrics, as can be seen in Figure 10, the Turner model produces predictions that are far superior to those obtained with the simplified model, regardless of the measure being considered. For this reason, despite its sophistication and the technicality of its implementation, the Turner model can be found in virtually all the predictive methods in the state of the art (described in the next section).

4.2 Tools

There is a large number of available tools for predicting the structure of RNA sequences. Among the most popular implementations is the `ViennaRNA` software suite [Lorenz et al., 2011]. It contains an extensive set of options and variations on folding/alignment, in addition to `Python` and `Perl` interfaces which can be easily integrated within an analysis pipeline. It is one of the most complete, and highly popular, tools in RNA bioinformatics due to these features. The `RNAstructure` package also offers many



(a) Sensitivity

(b) Positive Predictive Value (PPV)

(c) MCC $\approx \sqrt{\text{Sens.} \times \text{PPV}}$

Figure 10: Sensitivity, Positive Predictive Value (PPV), and Matthews Correlation Coefficient (MCC) of energy minimization-based prediction in two energy models: Simplified model based on base pairs (blue); and Turner model (orange).

Task	Section	Tool/command	Package/Ref.
Energy minimization	2.1	`RNAfold`	`ViennaRNA`
		`Fold`	`RNAStructure`
Suboptimal folding	2.2	`RNAsubopt`	`ViennaRNA`
		`AllSub`	`RNAStructure`
Comparative folding	2.3	`LocARNA`	`ViennaRNA`
		`dynalign`	`RNAStructure`
		`FoldAlign`	Sundfeld et al. [2015]
Partition function	3.1	`RNAfold -p`	`ViennaRNA`
		`partition`	`RNAStructure`
Statistical sampling	3.2	`RNAsubopt -p`	`ViennaRNA`
		`stochastic`	`RNAStructure`
Boltzmann probabilities	3.3	`RNAfold -p`	`ViennaRNA`
		`partition`	`RNAStructure`
Multiple folding	–	`RNAalifold`	`ViennaRNA`

Table 1: Reference implementations of the algorithms seen in this chapter (Turner energy model).

features, and provides a myriad of options, as well as a Java graphical interface [Reuter and Mathews, 2010]. Table 1 summarizes the main implementations of the algorithms presented in this chapter.

5 Bibliography

H M Berman, J Westbrook, Z Feng, G Gilliland, T N Bhat, H Weissig, I N Shindyalov, and P E Bourne. The protein data bank. *Nucleic acids research*, 28:235–242, January 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.235.

Ye Ding and Charles E Lawrence. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic acids research*, 31:7280–7301, December 2003. ISSN 1362-4962. doi: 10.1093/nar/gkg938.

Ye Ding, Chi Yu Chan, and Charles E Lawrence. RNA secondary structure prediction by centroids in a boltzmann weighted ensemble. *RNA (New York, N.Y.)*, 11:1157–1166, August 2005. ISSN 1355-8382. doi: 10.1261/rna.2500605.

Consortium ENCODE et al. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, 447(7146):799, 2007.

Sven Findeiß, Maja Etzel, Sebastian Will, Mario Mörl, and Peter F Stadler. Design

of artificial riboswitches as biosensors. *Sensors (Basel, Switzerland)*, 17(9):E1990, August 2017. ISSN 1424-8220. doi: 10.3390/s17091990.

O. Gotoh. An improved algorithm for matching biological sequences. *J Mol Biol*, 162:705–708, 1982.

Liang Huang and David Chiang. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 53–64, Vancouver, British Columbia, October 2005. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W05-1506`.

Ioanna Kalvari, Joanna Argasinska, Natalia Quinones-Olvera, Eric P Nawrocki, Elena Rivas, Sean R Eddy, Alex Bateman, Robert D Finn, and Anton I Petrov. Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Research*, 46(D1):D335–D342, 11 2017. ISSN 0305-1048. doi: 10.1093/nar/gkx1038. URL `https://doi.org/10.1093/nar/gkx1038`.

D. Lai, J. R. Proctor, and I. M. Meyer. On the importance of cotranscriptional RNA structure formation. *RNA*, 19(11):1461–1473, oct 2013. doi: 10.1261/rna.037390.112.

K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.

Ronny Lorenz, Stephan H Bernhart, Christian Höner Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. ViennaRNA package 2.0. *Algorithms for molecular biology : AMB*, 6:26, November 2011. ISSN 1748-7188. doi: 10.1186/1748-7188-6-26.

Xiang-Jun Lu, Harmen J. Bussemaker, and Wilma K. Olson. DSSR: an integrated software tool for dissecting the spatial structure of RNA. *Nucleic Acids Research*, page gkv716, jul 2015. doi: 10.1093/nar/gkv716.

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, jul 2004. doi: 10.1261/rna.7650904.

John S McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers: Original Research on Biomolecules*, 29(6-7):1105–1119, 1990.

Ulrike Mückstein, Hakim Tafer, Jörg Hackermüller, Stephan H Bernhart, Peter F Stadler, and Ivo L Hofacker. Thermodynamics of rna-rna binding. *Bioinformatics (Oxford, England)*, 22:1177–1182, May 2006. ISSN 1367-4803. doi: 10.1093/bioinformatics/btl024.

François Michel and Eric Westhof. Modelling of the three-dimensional architecture of group i catalytic introns based on comparative sequence analysis. *Journal of Molecular Biology*, 216(3):585–610, dec 1990. doi: 10.1016/0022-2836(90)90386-z.

Ruth Nussinov, George Pieczenik, Jerrold R Griggs, and Daniel J Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied mathematics*, 35(1):68–82,

1978.

Yann Ponty. Efficient sampling of RNA secondary structures from the Boltzmann ensemble of low-energy: The boustrophedon method. *Journal of Mathematical Biology*, 56(1-2):107–127, 2008. doi: 10.1007/s00285-007-0137-z. URL `https://hal.inria.fr/inria-00548863`.

Jessica S Reuter and David H Mathews. RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, 11(1), mar 2010. doi: 10.1186/1471-2105-11-129.

David. Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics*, 45(5):810–825, 1985. doi: 10.1137/0145048. URL `https://doi.org/10.1137/0145048`.

Daniel Sundfeld, Jakob H. Havgaard, Alba C. M. A. de Melo, and Jan Gorodkin. Foldalign 2.5: multithreaded implementation for pairwise structural RNA alignment. *Bioinformatics*, 32(8):1238–1240, dec 2015. doi: 10.1093/bioinformatics/btv748.

Douglas H Turner and David H Mathews. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic acids research*, 38(Database issue):D280–D282, January 2010. ISSN 1362-4962. doi: 10.1093/nar/gkp892.

Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, jan 1994. doi: 10.1089/cmb.1994.1.337.

Michael Waterman. Secondary structure of single-stranded nucleic acids. *Advances in Mathematics: Supplementary Studies*, 1:167–212, 1978.

Michael S. Waterman and Thomas H. Byers. A dynamic programming algorithm to find all solutions in a neighborhood of the optimum. *Mathematical Biosciences*, 77 (1-2):179–188, dec 1985. doi: 10.1016/0025-5564(85)90096-3.

Sebastian Will, Tejal Joshi, Ivo L Hofacker, Peter F Stadler, and Rolf Backofen. LocaRNA-p: accurate boundary prediction and improved detection of structural RNAs. *RNA (New York, N.Y.)*, 18:900–914, May 2012. ISSN 1469-9001. doi: 10.1261/rna.029041.111.

Sebastian Will, Christina Otto, Milad Miladi, Mathias Möhl, and Rolf Backofen. Sparse: quadratic time simultaneous alignment and folding of RNAs without sequence-based heuristics. *Bioinformatics (Oxford, England)*, 31:2489–2496, August 2015. ISSN 1367-4811. doi: 10.1093/bioinformatics/btv185.

Stefan Wuchty, Walter Fontana, Ivo L Hofacker, and Peter Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers: Original Research on Biomolecules*, 49(2):145–165, 1999.

Tianbing Xia, John SantaLucia Jr, Mark E Burkard, Ryszard Kierzek, Susan J Schroeder, Xiaoqi Jiao, Christopher Cox, and Douglas H Turner. Thermodynamic

parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson–Crick base pairs. *Biochemistry*, 37(42):14719–14735, 1998.

M Zuker. On finding all suboptimal foldings of an rna molecule. *Science (New York, N.Y.)*, 244:48–52, April 1989. ISSN 0036-8075. doi: 10.1126/science.2468181.

Michael Zuker and David Sankoff. RNA secondary structures and their prediction. *Bulletin of mathematical biology*, 46(4):591–621, 1984.

6 Bibliography

H M Berman, J Westbrook, Z Feng, G Gilliland, T N Bhat, H Weissig, I N Shindyalov, and P E Bourne. The protein data bank. *Nucleic acids research*, 28:235–242, January 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.235.

Ye Ding and Charles E Lawrence. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic acids research*, 31:7280–7301, December 2003. ISSN 1362-4962. doi: 10.1093/nar/gkg938.

Ye Ding, Chi Yu Chan, and Charles E Lawrence. RNA secondary structure prediction by centroids in a boltzmann weighted ensemble. *RNA (New York, N.Y.)*, 11:1157–1166, August 2005. ISSN 1355-8382. doi: 10.1261/rna.2500605.

Consortium ENCODE et al. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, 447(7146):799, 2007.

Sven Findeiß, Maja Etzel, Sebastian Will, Mario Mörl, and Peter F Stadler. Design of artificial riboswitches as biosensors. *Sensors (Basel, Switzerland)*, 17(9):E1990, August 2017. ISSN 1424-8220. doi: 10.3390/s17091990.

O. Gotoh. An improved algorithm for matching biological sequences. *J Mol Biol*, 162:705–708, 1982.

Liang Huang and David Chiang. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 53–64, Vancouver, British Columbia, October 2005. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W05-1506`.

Ioanna Kalvari, Joanna Argasinska, Natalia Quinones-Olvera, Eric P Nawrocki, Elena Rivas, Sean R Eddy, Alex Bateman, Robert D Finn, and Anton I Petrov. Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Research*, 46(D1):D335–D342, 11 2017. ISSN 0305-1048. doi: 10.1093/nar/gkx1038. URL `https://doi.org/10.1093/nar/gkx1038`.

D. Lai, J. R. Proctor, and I. M. Meyer. On the importance of cotranscriptional RNA structure formation. *RNA*, 19(11):1461–1473, oct 2013. doi: 10.1261/rna.037390.112.

K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.

Ronny Lorenz, Stephan H Bernhart, Christian Höner Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. ViennaRNA package 2.0. *Algorithms for molecular biology : AMB*, 6:26, November 2011. ISSN 1748-7188. doi: 10.1186/1748-7188-6-26.

Xiang-Jun Lu, Harmen J. Bussemaker, and Wilma K. Olson. DSSR: an integrated software tool for dissecting the spatial structure of RNA. *Nucleic Acids Research*, page gkv716, jul 2015. doi: 10.1093/nar/gkv716.

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, jul 2004. doi: 10.1261/rna.7650904.

John S McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers: Original Research on Biomolecules*, 29(6-7):1105–1119, 1990.

Ulrike Mückstein, Hakim Tafer, Jörg Hackermüller, Stephan H Bernhart, Peter F Stadler, and Ivo L Hofacker. Thermodynamics of rna-rna binding. *Bioinformatics (Oxford, England)*, 22:1177–1182, May 2006. ISSN 1367-4803. doi: 10.1093/bioinformatics/btl024.

François Michel and Eric Westhof. Modelling of the three-dimensional architecture of group i catalytic introns based on comparative sequence analysis. *Journal of Molecular Biology*, 216(3):585–610, dec 1990. doi: 10.1016/0022-2836(90)90386-z.

Ruth Nussinov, George Pieczenik, Jerrold R Griggs, and Daniel J Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied mathematics*, 35(1):68–82, 1978.

Yann Ponty. Efficient sampling of RNA secondary structures from the Boltzmann ensemble of low-energy: The boustrophedon method. *Journal of Mathematical Biology*, 56(1-2):107–127, 2008. doi: 10.1007/s00285-007-0137-z. URL `https://hal.inria.fr/inria-00548863`.

Jessica S Reuter and David H Mathews. RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, 11(1), mar 2010. doi: 10.1186/1471-2105-11-129.

David. Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics*, 45(5):810–825, 1985. doi: 10.1137/0145048. URL `https://doi.org/10.1137/0145048`.

Daniel Sundfeld, Jakob H. Havgaard, Alba C. M. A. de Melo, and Jan Gorodkin. Foldalign 2.5: multithreaded implementation for pairwise structural RNA alignment. *Bioinformatics*, 32(8):1238–1240, dec 2015. doi: 10.1093/bioinformatics/btv748.

Douglas H Turner and David H Mathews. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic acids research*, 38(Database issue):D280–D282, January 2010. ISSN 1362-4962. doi: 10.1093/nar/gkp892.

Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, jan 1994. doi: 10.1089/cmb.1994.1.337.

Michael Waterman. Secondary structure of single-stranded nucleic acids. *Advances in Mathematics: Supplementary Studies*, 1:167–212, 1978.

Michael S. Waterman and Thomas H. Byers. A dynamic programming algorithm to find all solutions in a neighborhood of the optimum. *Mathematical Biosciences*, 77

(1-2):179–188, dec 1985. doi: 10.1016/0025-5564(85)90096-3.

Sebastian Will, Tejal Joshi, Ivo L Hofacker, Peter F Stadler, and Rolf Backofen. LocaRNA-p: accurate boundary prediction and improved detection of structural RNAs. *RNA (New York, N.Y.)*, 18:900–914, May 2012. ISSN 1469-9001. doi: 10.1261/rna.029041.111.

Sebastian Will, Christina Otto, Milad Miladi, Mathias Möhl, and Rolf Backofen. Sparse: quadratic time simultaneous alignment and folding of RNAs without sequence-based heuristics. *Bioinformatics (Oxford, England)*, 31:2489–2496, August 2015. ISSN 1367-4811. doi: 10.1093/bioinformatics/btv185.

Stefan Wuchty, Walter Fontana, Ivo L Hofacker, and Peter Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers: Original Research on Biomolecules*, 49(2):145–165, 1999.

Tianbing Xia, John SantaLucia Jr, Mark E Burkard, Ryszard Kierzek, Susan J Schroeder, Xiaoqi Jiao, Christopher Cox, and Douglas H Turner. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson–Crick base pairs. *Biochemistry*, 37(42):14719–14735, 1998.

M Zuker. On finding all suboptimal foldings of an rna molecule. *Science (New York, N.Y.)*, 244:48–52, April 1989. ISSN 0036-8075. doi: 10.1126/science.2468181.

Michael Zuker and David Sankoff. RNA secondary structures and their prediction. *Bulletin of mathematical biology*, 46(4):591–621, 1984.