# Algorithmic Hardness of the Partition Function for Nucleic Acid Strands

**Gwendal Ducloz**
Hamilton Institute and Department of Computer Science, Maynooth University, Ireland
École Normale Supérieure de Lyon, France

**Ahmed Shalaby**
Hamilton Institute and Department of Computer Science, Maynooth University, Ireland

**Damien Woods**
Hamilton Institute and Department of Computer Science, Maynooth University, Ireland

## Abstract

To understand and engineer biological and artificial nucleic acid systems, algorithms are employed for prediction of secondary structures at thermodynamic equilibrium. Dynamic programming algorithms are used to compute the most favoured, or Minimum Free Energy (MFE), structure, and the Partition Function (PF) – a tool for assigning a probability to any structure. However, in some situations, such as when there are large numbers of strands, or pseudoknotted systems, NP-hardness results show that such algorithms are unlikely, but only for MFE. Curiously, algorithmic hardness results were not shown for PF, leaving two open questions on the complexity of PF for multiple strands and single strands with pseudoknots. The challenge is that while the MFE problem cares only about one, or a few structures, PF is a summation over the entire secondary structure space, giving theorists the vibe that computing PF should not only be as hard as MFE, but should be even harder.

We answer both questions. First, we show that computing PF is #P-hard for systems with an unbounded number of strands, answering a question of Condon Hajiaghayi, and Thachuk [DNA27]. Second, for even a single strand, but allowing pseudoknots, we find that PF is #P-hard. Our proof relies on a novel *magnification trick* that leads to a tightly-woven set of reductions between five key thermodynamic problems: MFE, PF, their decision versions, and #SSEL that counts structures of a given energy. Our reductions show these five problems are fundamentally related for any energy model amenable to magnification. That general classification clarifies the mathematical landscape of nucleic acid energy models and yields several open questions.

**2012 ACM Subject Classification** Theory of computation → Problems, reductions and completeness

**Keywords and phrases** Partition function, minimum free energy, nucleic acid, DNA, RNA, secondary structure, computational complexity, #P-hardness

**Digital Object Identifier** 10.4230/LIPIcs.DNA.31.1

**Related Version** *Full Version*: `https://arxiv.org/abs/2506.19756` [11]

**Funding** Work carried out while GD was on internship at Maynooth University. Supported by Science Foundation Ireland (SFI) under grant number 20/FFP-P/8843, European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 772766, Active-DNA project), and European Innovation Council (EIC), No 101115422, DISCO project. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union, ERC, EIC or SFI. Neither the European Union nor the granting authority can be held responsible for them.

**Acknowledgements** We thank Doan Dai Nguyen, Constantine Evans, Dave Doty, Sergiu Ivanov and Cai Wood for stimulating discussions.

© Gwendal Ducloz, Ahmed Shalaby, and Damien Woods;
licensed under Creative Commons License CC-BY 4.0

## 1 Introduction

The information encoding abilities of RNA is harnessed by biology to encode myriad complex behaviours, and both DNA and RNA have been used by scientists and engineers to create custom nanostructures and molecular computers. Consequently, predicting the structures formed by DNA and RNA strands is crucial both for understanding molecular biology and advancing molecular programming.

The *primary structure* of a DNA strand is simply a word over the alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$, with U instead of T for RNA. Bases bond in pairs, A-T and C-G, and a set of such pairings for one or more strands is called a *secondary structure*. Typically, the set of all secondary structures $\Omega$ has size exponential in the total number of bases. Assuming an energy model over secondary structures, each secondary structure $S$ has an associated real-valued free energy $\Delta G(S)$, where more negative means more favourable [27].

The Boltzman distribution is used to model the probability distribution of secondary structures at equilibrium [21, 10, 8]: the probability of $S$ is given by $p(S) = \frac{1}{Z}\mathrm{e}^{-\Delta G(S)/k_{\mathrm{B}}T}$, where $Z$ is a normalisation factor called the partition function (PF):

$$Z = \sum_{S \in \Omega} \mathrm{e}^{-\Delta G(S)/k_{\mathrm{B}}T} \tag{1}$$

Intuitively, $Z$ is an exponentially weighted sum of the free energies over $\Omega$, where $k_{\mathrm{B}}$ is Boltzmann's constant and $T$ is temperature in Kelvin. The algorithmic complexity of computing the PF is the main object of study in this paper.

Predicting the free energy of the most favoured secondary structure(s) at thermodynamic equilibrium is called the Minimum Free Energy (MFE) problem [32, 23, 33]. A third problem of interest, called #SSEL (or number of Secondary Structures at a specified Energy Level), asks a more fine-grained question [7]: Given a value $k \in \mathbb{R}$ how many secondary structures have free energy $k$? Here, we give a mathematical relation between all of these models (Figure 2) that is somewhat energy-model agnostic and settle the computational complexity of PF in two settings (Table 1).

**Table 1** Results on the computational complexity of MFE and PF. P: problem is solvable in time polynomial in $n$, the total number of DNA/RNA bases; NP-hard: as hard as any problem in nondeterministic polynomial time (NP); #P-hard: as hard as counting the accepting paths of an NP Turing machine. All positive results showing a problem in P hold for all 3 models studied here: base pair matching (BPM), base pair stacking (BPS), and nearest neighbour (NN). Negative, or hardness, results are proven in the simple BPM or BPS models and are conjectured [5] to hold in the more complex NN model. Specifically: Condon et al. [5] use the BPM model, and Lyngsø [19] uses the BPS model to prove the NP-completeness of the decision version of MFE. The two #P-hardness[3] results are contributions of this paper, along with the reductions of Figure 2.

| Structure | Num. Strands | MFE | PF |
|---|---|---|---|
| Unpseudoknotted | 1 | P [32, 23, 33] | P [21] |
| | Bounded ($\mathcal{O}(1)$) | P [26] | P [8] |
| | Unbounded ($\mathcal{O}(n)$) | NP-hard [5]; BPM model | **#P-hard** [Theorem 38]; BPM model |
| Pseudoknotted | 1 | NP-hard [1, 20, 19]; including BPS model | **#P-hard** [Theorem 33]; BPS model |

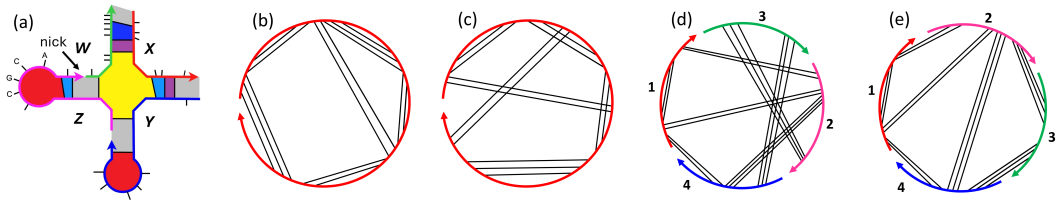


**Figure 1** The nearest neighbour, or Turner, model of DNA/RNA multistranded secondary structure. (a) One of the many possible secondary structures for four DNA (or RNA) strands $W, X, Y, Z$. Short black lines represent DNA bases (a few are shown $\ldots \mathrm{C}, \mathrm{G}, \mathrm{C}, \mathrm{A} \ldots$), and long lines represent base pairs (not to scale). Loops are colour-coded: stack (purple), hairpin (red), bulge (light blue), internal (dark blue), multiloop (yellow), external (grey). Black arrow: *nick*, the gap between two strands. (b–c) Polymer graphs[1] for two single-stranded secondary structures: (b) unpseudoknotted (no crossings), (c) pseudoknotted (crossings). (d) Polymer graph for a secondary structure $S$ over the strand set $\{1, 2, 3, 4\}$ with strand ordering 1324 showing crossings. (e) Simply by reordering the strands to 1234 gives a polymer graph without crossings, proving $S$ is unpseudoknotted.

## 1.1 Background and related work

**Efficient algorithms.** Decades ago, the relationship between secondary structure prediction and dynamic programming algorithms was well established. For a single strand of length $n$, dynamic programming techniques were used to solve the MFE and PF problems efficiently, meaning in polynomial time in the number of bases, but ignoring so-called pseudoknotted structures. Early algorithms were designed for simple energy models that count the number of base pairs, the base pair matching (BPM) model [32, 23]. Later algorithms accounted for more complex secondary structure features, including stacks, hairpin loops, internal loops and other features that comprise the *nearest nieghbour*, or *Turner*, model [21, 8], Figure 1, extensively used by practitioners [30, 17, 31].

**Hardness results for single strands and an open problem.** Somewhat frustratingly, dynamic programming algorithms for MFE prediction do not handle all secondary structures: as noted, pseudoknots (Definition 2) throw a spanner in the works. Pseudoknotted structures are ubiquitous in both biological RNA and DNA nanotech and computing systems, so why ignore them? In 2000, prediction in the presence of pseudoknots was shown to be NP-hard, even for a single strand and under simple energy models like the base pair stacking model (BPS; counts stacks) [1, 20, 19]. NP-hardness implies we are unlikely to see efficient algorithms for the full class of pseudoknots [13], although progress has been made on specific subclasses [10, 24, 1, 9, 4, 16]. Although it feels as though computing PF should be at least as hard as MFE, for reasons outlined below the complexity of PF in this setting (single strand, allowing pseudoknots) remained tantalisingly open.

**Hardness results for multiple strands and another open problem.** Recently, Condon, Hajiaghayi, and Thachuk [5] proved that computing MFE is also NP-hard in the unpseudoknotted BPM model when the number of strands is unbounded, meaning it scales with problem size. They left the complexity of PF in this setting as an open problem.

[1] A secondary structure can be represented as a polymer graph by ordering and depicting the directional ($5'$ to $3'$) strands around the circumference of a circle, with edges along the circumference representing adjacent bases, and straight line edges connecting paired bases. Each such ordering of $c$ strands is a circular permutation of the strands, and there are $(c-1)!$ possible orderings [5, 26, 8].

**Results on #SSEL.** The counting problem #SSEL is also known in the literature as the density of states problem [18]. Efficient dynamic programming algorithms can be adapted to solve a version of this problem where energies fall into discrete bins (ranges), for unpseudoknotted secondary structures [6]. In addition, probabilistic heuristic methods have been proposed to estimate its value with good accuracy [18]. These heuristics originate from statistical physics, particularly from the study of the Ising model. Interestingly, this model draws a useful parallel, as computing its partition function is known to be either in P or #P-complete, depending on the parameters [14].

**Indirect evidence that PF might be harder than MFE.** In 2007, Dirks et al. [8] gave a polynomial time dynamic programming algorithm for PF for multiple, but $\mathcal{O}(1)$, strands in the NN model [8]. Their paper includes a definitional contribution that extends the single-strand NN model to multiple strands by including both a strand association penalty and an entropic penalty for rotational symmetry of multi-stranded secondary structures. They observe if the model is permitted to ignore rotational symmetry, any purely dynamic programming algorithm for PF can be translated into an algorithm for MFE using tropical $(\min, +)$ algebra instead of the classic $(+, \cdot)$ algebra. But going the other way is not so obvious: translating MFE algorithms into PF algorithms is challenging due to the risk of overcounting secondary structures, by translating to an overly naive algorithm. This observation provides some intuition that PF might be harder than MFE.

Recently Shalaby and Woods [26] gave an efficient algorithm for computing MFE in the same setting as Dirks et al. [8] ($\mathcal{O}(1)$ strands, unpseudoknotted). For roughly two decades, this setting had supported an efficient algorithm for PF but none for MFE due to rotational symmetry complications. The rareness of the situation and its positive resolution bolstered our intuition PF is not easier than MFE.

**A note about energy models.** Efficient algorithms seem to work across a range of energy models (BPM, BPS, NN),[2] but NP-hardness results have merely been shown for MFE in simple energy models: BPM and BPS. As Condon, Hajiaghayi, and Thachuk observe [5], it seems unlikely MFE would become easier in more sophisticated models like NN.

## 1.2 Contributions

We prove hardness results on the complexity of PF in two settings (Table 1), and provide tools for thinking about the complexity of thermodynamic prediction problems (Figure 2).

Section 2 contains definitions of secondary structures, energy models (BPM, BPS and NN), and thermodynamic problems (MFE, PF, #SSEL, and decision problems dMFE and dPF.)

Our first main contribution is to provide a map of reductions between all of these problems, illustrated in Figure 2 with proofs in Section 3. Some of these reductions are rather straight-forward (grey arrows in Figure 2), but the others (red arrows) make use of a new proof strategy we call the energy *magnification trick*. In Section 3.5 we define a property of an energy model called *PF-polynomially magnifiable* (Definition 25) which means that the energy model has a polynomial time magnification adaptable PF algorithm. That general

[2] The BPS model is a special case of the NN model, so any algorithm in the NN model can be modified for the BPS model by easily ignoring all loops except stacks and setting $\Delta G^{\text{stack}} = -1$.

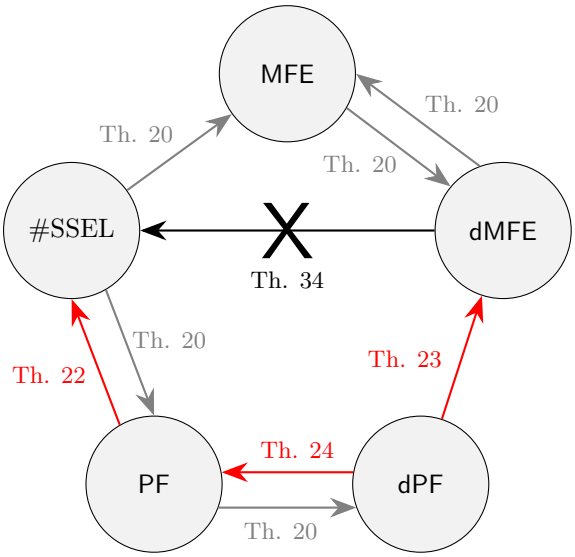


**Figure 2** Reduction map between the five main problems (Section 2.3) studied in this paper, results shown in Section 3. An arrow from problem A to B means that if there exists an algorithm for A, it can be called to efficiently solve B, or in other words that there is a polynomial-time Turing reduction from B to A. Red arrows use a magnification of the energy model, grey arrows do not. A crossed arrow signifies that no such reduction exists unless $\#P \subseteq P^{NP}$, which means the collapse of polynomial hierarchy [2] at level 2. The latter implies the non-existence of arrows from dMFE to dPF and from MFE to #SSEL.

property yields Corollary 26, that all 5 problems in Figure 2 are in P (or FP) whenever the underlying energy model is PF-polynomially magnifiable. Section 3.5 has a full discussion about PF-polynomially magnifiable energy models and magnification adaptable algorithms.

Our second main contribution is to answer two open problems on the complexity of PF: we show that PF is #P-hard[3] in (a) the single-stranded case, i.e. with pseudoknots under the BPM model, and (b) in the multi-stranded case even without pseudoknots (Table 1) under the BPS model. These results are proven in Sections 4 and 5, and leverage the reduction map (Figure 2, Section 3). Since dMFE is NP-complete [5, 19], these #P-hardness results provide a strong result, showing that PF is strictly harder than dMFE in these two situations, unless $\#P \subseteq P^{NP}$, which implies the collapse of the polynomial hierarchy [2] at level 2.

In addition to showing the #P-hardness lowerbound on PF, the reductions in Section 3 show that PF and #SSEL have equivalent complexity in the sense that they are polynomial-time Turing reducible [2] to each other, providing an upperbound on the complexity of PF.

Although our #P-hardness results are shown for the BPM and BPS models, our reductions apply to the NN model, and Appendix A has some analysis of the NN model.

[3] A first upperbound of all five problems in Figure 2 is the exponential time class EXP, as we can solve any problem just by going though all distinct secondary structures. Therefore, the interesting question is about finding better complexity lowerbounds and upperbounds. The complexity class #P, introduced by Leslie Valiant [29], is the class of problems where the goal is to count the number of solutions, with each solution having a polynomial-sized certificate.

## 1.3 Future work

1. MFE hardness in two settings is still an open question: Is MFE NP-hard for NN single-strand pseudoknotted, or NN $\mathcal{O}(n)$-strands unpseudoknotted? If so, can the reduction map be leveraged to show hardness results for PF and #SSEL in the NN model?
2. Using the reduction map (Figure 2) for positive results: Is the NN model PF-polynomially magnifiable (see Definition 25)? (The existing algorithm by Dirks et al. [8] does not handle rotational symmetry in a magnification adaptable way.) A positive answer, and the reduction map would immediately imply a polynomial time algorithm for #SSEL. (There already is a polynomial time algorithm for MFE [26], and hence dMFE.)
3. Figure 2 has arrows of two colours: red that means the reduction uses our magnification trick, and grey does not. Can all red arrows be made grey? This involves replacing our magnification trick with a completely different strategy.
4. In Note 13, we assume in the NN model that loops are specified using at most logarithmic precision. Logarithmic precision seems reasonable from a physical perspective, since parameters to the NN free energy model have uncertainty after only a few decimal places. Mathematically, some loop free energies, namely, hairpin, interior and bulge loops, are a logarithmic function of their size and thus may be irrational. Hence we ask: Can this logarithmic precision assumption be dropped?
5. We studied the MFE problem, which aims to compute the minimum free energy value. A natural variant is to ask for its corresponding secondary structures. How does this version relate to our five other problems?

# 2 Definitions

First, we need to review some basic terminology to formulate our main problems in an easy and precise way. We set the scene, in Section 2.1, by the mathematical definitions for single-stranded nucleic acid systems before extending them to the multi-stranded case. In Section 2.2, we provide a brief review of existing energy models and some abstract properties of these models that play a significant role in our reductions. Finally, in Section 2.3, we provide the formal definitions of the main problems of interest in this work.

## 2.1 Single-stranded and multi-stranded nucleic acid systems

Formally, a DNA strand $s$ is a word over the alphabet of DNA *bases* $\{\mathrm{A}, \mathrm{T}, \mathrm{G}, \mathrm{C}\}$, indexed from 1 to $|s| = n$, where $n$ denotes the number of bases of $s$. Hydrogen bonds, or base pairs, can form between complementary bases, namely C–G and A–T, and formally such a base pair is written as a tuple $(i, j)$, where $i < j$, of strand indices (indexing from 1).

▶ **Definition 1** (Single-stranded secondary structure $S$)**.** For any DNA strand $s$, a secondary structure $S$ of $s$ is a set of base pairs, such that each base appears in at most one pair, i.e. if $(i, j) \in S$ and $(k, l) \in S$ then $i, j, k, l$ are all distinct or $(i, j) = (k, l)$.

▶ **Definition 2** (Unpseudoknotted single-stranded secondary structure)**.** A secondary structure $S$ of a strand $s$ is unpseudoknotted if for any two base pairs $(i, j)$ and $(k, l) \in S$, $i < k < j$ if and only if $i < l < j$. Otherwise, we call $S$ pseudoknotted, see Figure 1.

▶ Remark 3. The maximum number of base pairs in any secondary structure $S$ of strand $s$ is $\lfloor n/2 \rfloor$. Hence, $S$ is representable in space polynomial in $n$, and verifying whether $S$ is a valid secondary structure is computable in polynomial time.

We extend these definitions to a multiset s of $c \in \{1, 2, 3, \ldots\}$ interacting nucleic acid strands, with $c = 1$ is called **single-stranded** and $c > 1$ **multi-stranded**. Each strand has a unique identifier $t \in \{1, \ldots, c\}$ [8], and a base $i_t$ has a strand identifier $t$ and an index $1 \leq i \leq l_t$, where $l_t$ is the length of the $t^{\text{th}}$ strand. In this case, $n$ is the total number of bases: $n = \sum_{1 \leq t \leq c} l_t$.

▶ **Definition 4** (Multi-stranded secondary structure $S$)**.** For $c \in \mathbb{N}$ interacting strands, a secondary structure $S$ is a set of base pairs such that each base appears in at most one pair, i.e. if $(i_n, j_m) \in S$ and $(k_q, l_r) \in S$ then $i_n, j_m, k_q, l_r$ are all distinct or $(i_n, j_m) = (k_q, l_r)$.

▶ **Definition 5** (Unpseudoknotted multi-stranded secondary structure)**.** For $c \in \mathbb{N}$ interacting strands, we call a secondary structure $S$ unpseudoknotted if there exists at least one ordering $\pi$ of the $c$ strands such that if we consider the $c$ strands in a row as forming one single long strand, with lexicographically ordered bases, then $S$ is unpseudoknotted according to Definition 2, see Figure 1.

## 2.2 Energy models

We are interested in the computational complexity and relationships between five problems (Section 2.3) with an energy model as part of their input. First, we review three important energy models from the literature. Appendix A provides an analysis of the set of *candidate energy levels* for each model.

▶ **Definition 6** (Energy model)**.** An **energy model** $\mathcal{M}$ defines a free energy function $\Delta G^{\mathcal{M}}$, such that $\Delta G^{\mathcal{M}} : \Omega_s \times \mathbb{R}^+ \to \mathbb{R}$, assigns a real value $\Delta G^{\mathcal{M}}(S, T)$ to any secondary structure $S$ of strand $s$, given a temperature $T$ (in Kelvin), where $\Omega_s$ is the set of all secondary structures (under interest) of $s$.

▶ Note 7 (Magnification of an energy model). For a positive real number $\alpha \in \mathbb{R}^+$ the notation $\alpha \cdot \Delta G^{\mathcal{M}}$ simply means to multiply the free energy function $\Delta G^{\mathcal{M}}(S, T)$ by $\alpha$. Whenever we apply magnification in this paper, we do so uniformly over all secondary structures $S$, which does not change their relative free energy ordering nor distribution per free energy level. This magnification is simple to compute for any given $S$ (just a multiplication), however it may not be obvious how to modify a sophisticated PF or MFE algorithm to be magnification compatible, as discussed in Section 3.5.

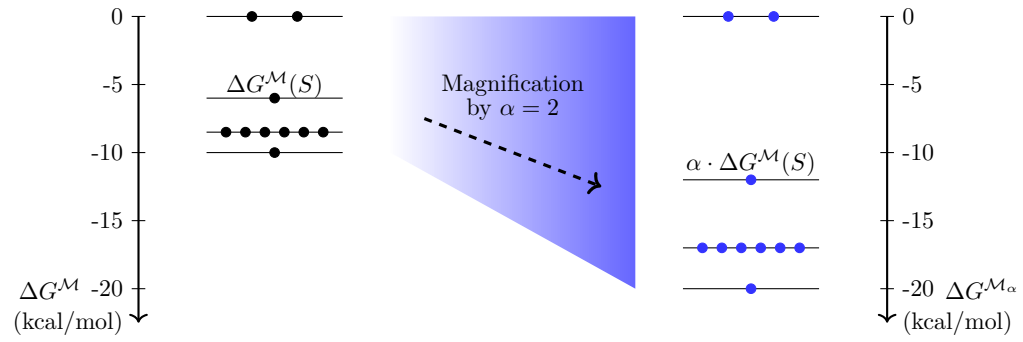


**Figure 3** Illustration of the magnification process used in Theorem 22 and Theorem 23. Each node is the free energy of a secondary structure $S$, with nodes on the left being $\Delta G^{\mathcal{M}}(S)$ and the right being $\Delta G^{\mathcal{M}_\alpha}(S) = \alpha \cdot \Delta G^{\mathcal{M}}(S)$. Magnification increases the absolute value of all energy levels, without changing the distribution of secondary structures per energy level.

▶ **Definition 8** (Base pair matching (BPM) model)**.** In the BPM model, the free energy of any secondary structure $S$, denoted by $\Delta G^{\mathrm{BPM}}(S)$, is the number of base pairs formed in $S$, such that each is weighted $-1$, hence: $\Delta G^{\mathrm{BPM}}(S) = -|S|$.

Despite the simplicity of the BPM model [22], it is still powerful enough to prove hardness results. For example, Condon, Hajiaghayi, and Thachuk [5] used it to prove the NP-hardness of MFE prediction of an unbounded set of strands in the unpseudoknotted case. In 2004, Lyngsø [19] introduced another energy model, called the base pair stacking (BPS) model, and proved NP-hardness of MFE for single-stranded pseudoknotted systems with stacks.

▶ **Definition 9** (Base pair stacking (BPS) model)**.** The number of base pair stackings (BPS) of any secondary structure $S$ is defined as $\mathrm{BPS}(S) = |\{(i,j) \in S \mid (i+1, j-1) \in S\}|$. In the BPS model, the free energy of a secondary structure $S$ is $\Delta G^{\mathrm{BPS}}(S) = -\mathrm{BPS}(S)$.

▶ Note 10. We say that an energy model $\mathcal{M}$ is **temperature independent** if its energy function $\Delta G^{\mathcal{M}}$ is not a function of temperature. Hence, $\Delta G^{\mathcal{M}}(S,T) = \Delta G^{\mathcal{M}}(S)$ for any $T$. The BPM and BPS models are temperature independent, and the NN model (below) is not.

A word of caution. From classical thermodynamics [27, 8]: free energies are typically of the form $\Delta G = \Delta H - T\Delta S$ meaning they are a function of temperature $T$ and can be decomposed into enthalpic ($\Delta H$) and entropic ($\Delta S$) contributions. Whether DNA/RNA binding occurs is strongly temperature dependent, which is why free energies are too. Thus, temperature independent energy models are not good representations of typical temperature-varying scenarios, however they are a useful vehicle to show computational complexity results in a fixed-temperature setting.

Beyond temperature dependence, the BPM and BPS models do not account for a number of features of DNA and RNA that provide significant free energy contributions: single-stranded regions and global symmetry. This motivates our third, most realistic, energy model which is called the nearest neighbour (NN) model:

▶ **Definition 11** (Nearest neighbour (NN) model)**.** Let $S$ be an unpseudoknotted connected[4] secondary structure over a multiset $s$ of $c \geq 1$ strands. $S$ s decomposed into a multiset of loops, denoted $\mathrm{loops}(S,s)$, each being one of the types: hairpin, interior, exterior, stack, bulge, and multiloop, as described in Figure 1. Then, the free energy of $S$ is the sum:

$$\Delta G^{\mathrm{NN}}(S) = \sum_{l \in \mathrm{loops}(S,s)} \Delta G(l) + (c-1)\Delta G^{\mathrm{assoc}} + k_{\mathrm{B}}T \log R$$

where $\Delta G : \mathrm{loops}(S,s) \to \mathbb{R}$ gives a free energy for each loop [25], $\Delta G^{\mathrm{assoc}} \in \mathbb{R}$ is an association penalty applied $c-1$ times for a complex of $c$ strands [8], and $R$ is the maximum degree of rotational symmetry [26, 8] of $S$, details follow.

A few comments are warranted on Definition 11. At fixed temperature, salt concentration, etc., the loop free energy $\Delta G(l)$ for a stack $l$ is simply a function of the stack's arrangement of its four constituent DNA/RNA bases [25]. For loops $l$ with single-stranded regions (e.g. interior, hairpin) $\Delta G(l)$ is a logarithmic function of loop length [8, 25], although for multiloops a linear approximation is used to facilitate dynamic programming [8]. We write the multiset of loops as a function of both the secondary structure $S$ and strand $s$ to emphasise that both base pair indices, and base identities are used to define loops. In this work, we

[4] In the NN model, unlike the BPM and BPS models, multi-stranded secondary structures must be connected [8], meaning the polymer graph of a secondary structure is connected.

consider $\Delta G^{\text{assoc}}$ to be a constant, however see [8] for more details, including temperature and water-molarity dependence. Versions of the NN model are implemented in software suites like `NUPACK` [8, 12], `ViennaRNA` [17], and `mfold` [31]. For more analysis of the NN model, see Appendix A.

▶ **Definition 12** (Set of candidate energy levels)**.** Given an energy model $\mathcal{M}$ and strand $s$ (or a set of strands $s$), a **set of candidate energy levels** $\mathcal{G}_s^{\mathcal{M}}$ is a finite superset of the energies of all secondary structures of $s$, in other words $\{\Delta G^{\mathcal{M}}(S) \mid S \in \Omega_s\} \subseteq \mathcal{G}_s^{\mathcal{M}}$.

▶ Note 13. $\mathcal{G}_s^{\mathcal{M}}$ is defined as a *superset* so that it is easily computable: specifically, computing $\mathcal{G}_s^{\mathcal{M}}$ does not require computing the MFE.

In Appendix A we show that there are sets of candidate energy levels of polynomial size and computable in polynomial time for all three models studied. For BPM and BPS the proofs are straightforward, but for NN we rely on an assumption that individual loop free energies are specified using logarithmic precision, by which we mean they can be written down as a rational number using $O(\log n)$ digits in units of kcal/mol. Physically, this is a reasonable assumption since measuring such free energies beyond a few decimal places is likely quite challenging. Mathematically, the assumption is not a trivial one, since hairpin, interior and bulge loops involve taking logarithms of natural numbers resulting in irrational free energies. However, computationally the assumption seems reasonable as any implemented algorithm will have finite precision.

## 2.3 Definitions of problems: MFE, PF, dMFE, dPF and #SSEL

In this section, we define the main five problems for which we establish computational complexity relationships. For convenience, we present the definitions for a single strand – the multi-stranded case is defined similarly, but the strand $s$ is replaced by a multiset of strands, and $n$ denotes the sum of strand lengths, or total number of bases, of the system.

▶ **Definition 14** (MFE; a function problem)**.**
**Input:** Nucleic acid strand $s$ of length $n$, a temperature $T \geq 0$, and an energy model $\mathcal{M}$.
**Output:** The minimum free energy $\text{MFE}(s, T, \mathcal{M}) = \min\{\Delta G^{\mathcal{M}}(S, T) \mid S \in \Omega_s\}$, where $\Omega_s$ is the set of all secondary structures (under interest) of $s$.

▶ **Definition 15** (PF; a function problem)**.**
**Input:** Nucleic acid strand $s$ of length $n$, a temperature $T \geq 0$, and an energy model $\mathcal{M}$.
**Output:** The partition function $\text{PF}(s, T, \mathcal{M}) = \sum_{S \in \Omega_s} e^{-\Delta G^{\mathcal{M}}(S,T)/k_{\text{B}}T}$, where $\Omega_s$ is the set of all secondary structures (under interest) of $s$.

▶ **Definition 16** (dMFE; a decision problem)**.**
**Input:** Nucleic acid strand $s$ of length $n$, a temperature $T \geq 0$, an energy model $\mathcal{M}$ and a value $k \in \mathbb{R}$.
**Output:** Is $\text{MFE}(s, T, \mathcal{M}) \leq k$?

▶ **Definition 17** (dPF; a decision problem)**.**
**Input:** Nucleic acid strand $s$ of length $n$, a temperature $T \geq 0$, an energy model $\mathcal{M}$, and a value $k \in \mathbb{R}$.
**Output:** Is $\text{PF}(s, T, \mathcal{M}) \geq k$?

▶ **Definition 18** (#SSEL; a counting problem)**.**
**Input:** Nucleic acid strand $s$ of length $n$, a temperature $T \geq 0$, an energy model $\mathcal{M}$, and a value $k \in \mathbb{R}$.
**Output:** $\#\text{SSEL}(s, T, \mathcal{M}, k)$: number of secondary structures $S$ of $s$ such that $\Delta G^{\mathcal{M}}(S) = k$.

▶ **Definition 19** (Polynomial-time Turing reduction)**.** A polynomial-time Turing reduction [2] from a problem $A$ to a problem $B$ is an algorithm that solves problem A using a polynomial number of calls to a subroutine for problem B, and polynomial time outside of those subroutine calls.

## 3 Reductions between the computational thermodynamic problems

Figure 2 illustrates most of the results of this section. For convenience, all proofs are written for the single-stranded case. However, all results hold in the multi-stranded case, simply by replacing the unique input strand by a set of multiple strands in the proofs.

### 3.1 Straightforward reductions

In this subsection, we prove all straightforward reductions (gray colored) in our reduction map in Figure 2. In the last three reductions, we use our assumption in Note 13 about the set of candidate energy levels.

▶ **Theorem 20.** *There exist the following polynomial-time Turing reductions:* dMFE *to* MFE*,* dPF *to* PF*,* MFE *to* dMFE*,* MFE *to* #SSEL*, and* PF *to* #SSEL*.*

**Proof.**

1. **Reduction from dMFE to MFE:** Let $(s, T, \mathcal{M}, k)$ be an input for the dMFE problem. After a single call to the $\mathrm{MFE}(s, T, \mathcal{M})$ function, simply computing the Boolean value $(\mathrm{MFE}(s, T, \mathcal{M}) \leq k)$ gives the answer to the dMFE problem.
2. **Reduction from dPF to PF:** Let $(s, T, \mathcal{M}, k)$ be an input for the dPF problem. Similarly, if the partition function $\mathrm{PF}(s, T, \mathcal{M})$ is known, then the Boolean value $(\mathrm{PF}(s, T, \mathcal{M}) \geq k)$ is the answer to the dMFE problem.
3. **Turing Reduction from MFE to dMFE:** Let $(s, T, \mathcal{M})$ be an input for the MFE problem. We know that $\mathrm{MFE}(s, T, \mathcal{M}) = g_j$ for some $1 \leq j \leq |\mathcal{G}_s^{\mathcal{M}}|$ by definition of $\mathcal{G}_s^{\mathcal{M}}$. Determining the MFE is equivalent to determining the integer $j$, which can be achieved through a binary search over $\mathcal{G}_s^{\mathcal{M}}$ thanks to the dMFE oracle. The complexity of this search is $\mathcal{O}(\log(|\mathcal{G}_s^{\mathcal{M}}|)) \in \mathcal{O}(\mathrm{poly}(n))$, as $\mathcal{G}_s^{\mathcal{M}}$ is of polynomial size. This binary search defines a Turing reduction from MFE to dMFE.
4. **Turing Reduction from MFE to** #SSEL: Let $(s, T, \mathcal{M})$ be an input for the MFE problem, to determine $\mathrm{MFE}(s, T, \mathcal{M})$, we linearly search for $j = \max_{1 \leq i \leq |\mathcal{G}_s^{\mathcal{M}}|}\{i \mid \#\mathrm{SSEL}(s, T, \mathcal{M}, g_i) \neq 0\}$ (i.e. the maximal such $i$ gives the MFE due to how the indices are ordered). This search requires only a polynomial number of calls to the #SSEL oracle, giving a polynomial time Turing reduction from MFE to #SSEL.
5. **Turing Reduction from PF to** #SSEL**:** Let $(s, T, \mathcal{M})$ be an input for the PF problem. The partition function $\mathrm{PF}(s, T, \mathcal{M}) = \sum_{1 \leq i \leq |\mathcal{G}_s^{\mathcal{M}}|} \#\mathrm{SSEL}(s, T, \mathcal{M}, g_i) e^{-g_i / k_{\mathrm{B}} T}$. This computation is computable in polynomial time as it requires a polynomial number of calls to the #SSEL oracle, hence defining a Turing reduction from PF to #SSEL. ◀

### 3.2 Polynomial-time Turing Reduction from #SSEL to PF

In this section, we prove one of the red-arrow reductions in Figure 2. The number of secondary structures of a strand $s$, denoted $\#\mathrm{SecStruct}(s)$, plays an important role in our reductions. Note that $\#\mathrm{SecStruct}(s)$ includes pseudoknotted and unpseudoknotted structures. The proof of the following lemma is in Appendix B.1.

▶ **Lemma 21.** *For any strand $s$ of size $n > 2$, $\#\mathrm{SecStruct}(s) < n!$.*

▶ **Theorem 22.** *There exists a polynomial-time Turing reduction from $\#\mathrm{SSEL}$ to $\mathrm{PF}$.*

**Proof.** For notational convenience, instead of $\#\mathrm{SSEL}(s, T, \mathcal{M}, g_i)$ we write $\#\mathrm{SSEL}(g_i)$ to denote the number of secondary structures with free energy $g_i$, and $\beta = 1/k_{\mathrm{B}}T$. Let $N = |\mathcal{G}_s^{\mathcal{M}}|$ denote the size of the set of candidate energy levels, see Definition 12 and Note 13.

Algorithm 1 gives a polynomial time Turing reduction from $\#\mathrm{SSEL}$ to $\mathrm{PF}$, for the inputs $s, T, \mathcal{M}, g_k$, where the main idea behind this algorithm is the following:

- We use our *magnification trick* to magnify the distances between energy levels: this idea imagines another energy model $\mathcal{M}_j$, where $\Delta G^{\mathcal{M}_j} = j \cdot \Delta G^{\mathcal{M}}$ (Definition 6 defines $\Delta G^{\mathcal{M}}$) which we assume a PF oracle can handle (see Section 3.5 for details).
- We compute PF using a call to the oracle $\mathrm{PF}(s, T, \mathcal{M}_j)$ with energy model $\mathcal{M}_j$.
- We do this magnification $N$ times with different magnification factors $1 \leq j \leq N$.
- We end up with a system of linear equations that has a unique solution which is the number of secondary structures per energy levels, outputting the correct one $\#\mathrm{SSEL}(g_k)$.

**Algorithm 1** Computing $\#\mathrm{SSEL}$ by calling the oracle $\mathrm{PF}(s, T, \mathcal{M})$ for PF.

**Input:** $s, T, \mathcal{M}, g_k$

1: Compute a set of candidate energy levels $\mathcal{G}_s^{\mathcal{M}} = \{g_1, \cdots, g_N\}$ ▷ see Appendix A

2: **for** $j \leftarrow 1$ to $N$ **do**

3: Let $b_j = \mathrm{PF}(s, T, \mathcal{M}_j)$, where $\Delta G^{\mathcal{M}_j} = j \cdot \Delta G^{\mathcal{M}}$ ▷ see Note 7

4: **end for**

5: Solve the following system of linear equations with $N$ unknowns $\{\#\mathrm{SSEL}(g_i),\ 1 \leq i \leq N\}$ and with $\beta = 1/k_{\mathrm{B}}T$:

$$\begin{array}{ccccccc}
\#\mathrm{SSEL}(g_1)e^{-\beta g_1} & + & \#\mathrm{SSEL}(g_2)e^{-\beta g_2} & + \cdots + & \#\mathrm{SSEL}(g_N)e^{-\beta g_N} & = & b_1 \\
\#\mathrm{SSEL}(g_1)(e^{-\beta g_1})^2 & + & \#\mathrm{SSEL}(g_2)(e^{-\beta g_2})^2 & + \cdots + & \#\mathrm{SSEL}(g_N)(e^{-\beta g_N})^2 & = & b_2 \\
\vdots & & \vdots & \ddots & \vdots & & \vdots \\
\#\mathrm{SSEL}(g_1)(e^{-\beta g_1})^N & + & \#\mathrm{SSEL}(g_2)(e^{-\beta g_2})^N & + \cdots + & \#\mathrm{SSEL}(g_N)(e^{-\beta g_N})^N & = & b_N
\end{array}$$

6: **Return:** $\#\mathrm{SSEL}(g_k)$

**Correctness.** An algorithm to compute the set $\mathcal{G}_s^{\mathcal{M}}$ is given in Appendix A. First note that the partition function can be partitioned by energy levels $g_i$ to give the form: $\mathrm{PF}(s, T, \mathcal{M}) = \sum_{i=1}^{N} \#\mathrm{SSEL}(g_i)e^{-\beta \cdot g_i}$. Hence, under magnification $j$ (see Note 7), the general expression for all $b_j$ is:

$$b_j = \mathrm{PF}(s, T, \mathcal{M}_j) = \sum_{i=1}^{N} \#\mathrm{SSEL}(g_i)e^{-\beta \cdot j \cdot g_i} = \sum_{i=1}^{N} \#\mathrm{SSEL}(g_i)(e^{-\beta \cdot g_i})^j$$

which is the form in Algorithm 1. Algorithm 1 solves this system of linear equations; to see this we write it in the standard matrix form $\mathbf{Ax} = \mathbf{b}$, where:

$$\mathbf{A} = \begin{pmatrix} e^{-\beta \cdot g_1} & e^{-\beta \cdot g_2} & \cdots & e^{-\beta \cdot g_N} \\ (e^{-\beta \cdot g_1})^2 & (e^{-\beta \cdot g_2})^2 & \cdots & (e^{-\beta \cdot g_N})^2 \\ \vdots & \vdots & \ddots & \vdots \\ (e^{-\beta \cdot g_1})^N & (e^{-\beta \cdot g_2})^N & \cdots & (e^{-\beta \cdot g_N})^N \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} \#\mathrm{SSEL}(g_1) \\ \#\mathrm{SSEL}(g_2) \\ \vdots \\ \#\mathrm{SSEL}(g_N) \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}$$

Since $\mathbf{A}$ is a Vandermonde matrix [15], and $e^{-\beta \cdot g_i} \neq e^{-\beta \cdot g_j}$ if $i \neq j$, then matrix $\mathbf{A}$ is non-singular. Therefore, the system of equations $\mathbf{Ax} = \mathbf{b}$ has a unique solution. That solution is the list $[\#\text{SSEL}(g_1), \#\text{SSEL}(g_k), \ldots, \#\text{SSEL}(g_N)]$, the $k$th entry being $g_k$, which is returned.

**Time analysis of Algorithm 1.** By the results in Appendix A, a candidate set $\mathcal{G}_s^{\mathcal{M}}$ is computed in polynomial time. $N = \text{poly}(n)$, therefore, Algorithm 1 calls the PF oracle $\text{PF}(s, T, \mathcal{M}_j)$ a polynomial number of times. Moreover, each call has polynomial input size. Solving the resulting system of $N$ linear equations can be achieved in $\text{poly}(N)$ time by Gaussian elimination, since the inputs of the system are stored in polynomial space:

- Matrix $\mathbf{A}$: the largest entry of $\mathbf{A}$ has size $\log((e^{-\beta \cdot g_N})^N) = N \log(e^{-\beta \cdot g_N}) = \text{poly}(n)$, since $g_i = \text{poly}(N) = \text{poly}(n)$, Note 13.
- Vector $\mathbf{b}$: the largest entry of $\mathbf{b}$ has size

$$\begin{aligned}
\log(b_N) &= \log(\#\text{SSEL}(g_1)(e^{-\beta g_1})^N + \cdots + \#\text{SSEL}(g_N)(e^{-\beta g_N})^N) \\
&\leq \log(N \#\text{SecStruct(s)}(e^{-\beta g_N})^N) \\
&\leq \log(N) + \log(\#\text{SecStruct(s)}) + N \log(e^{-\beta g_N}) \\
&\leq \text{poly}(n) \qquad \text{As } \#\text{SecStruct}(s) < n! \text{ by Lemma 21.}
\end{aligned}$$

This shows the existence of a polynomial-time Turing reduction from #SSEL to PF. ◀

## 3.3 Reduction from dMFE to dPF

▶ **Theorem 23.** *There exists a polynomial-time Turing reduction from* dMFE *to* dPF.

**Proof.** We denote the minimal step between any two energy levels by $\delta$, Note 13, and $\beta = 1/(k_{\text{B}}T)$. Algorithm 2 gives a polynomial time Turing reduction from dMFE to dPF, where the main idea behind this algorithm is the following:

- We use our magnification trick to make a **huge** magnification of energy levels.
- This magnification is carefully designed so that the contribution to the PF of exactly one secondary structure, the MFE one, overwhelms the contribution to PF of others.
- Our construction guarantees this by handling the worst case scenario:
  1. Only one secondary structure is at the MFE level.
  2. $\#\text{SecStruct}(s)$ secondary structures, see Lemma 21, belong to the closest energy level to the MFE level, which is MFE + $\delta$.
- This huge separation of the contribution of secondary structures is achieved due to the exponential nature of the PF. That, in turn, solves dMFE problem with a dPF oracle.

**Algorithm 2** Solve dMFE calling an oracle for dPF.

**Input:** $s, T, \mathcal{M}, k$
1: Compute a set of candidate energy levels $\mathcal{G}_s^{\mathcal{M}}$ ▷ see Appendix A
2: Let $x = \max\{g \in \mathcal{G}_s^{\mathcal{M}} \mid g \leq k\}$ ▷ $x$ is the max value from $\mathcal{G}_s^{\mathcal{M}}$ that is $\leq k$.
3: Let $\Delta G^{\mathcal{M}'} = (\log(n!)/(\beta\delta)) \cdot \Delta G^{\mathcal{M}}$
4: Let $k' = e^{-\log(n!) \cdot x/\delta} = (n!)^{-x/\delta}$
**Return:** $\text{dPF}(s, T, \mathcal{M}', k')$

**Correctness.** We need to prove the following claim: $\text{dMFE}(s, T, \mathcal{M}, k) \Leftrightarrow \text{dPF}(s, T, \mathcal{M}', k')$, which is equivalent to $\text{MFE}(s, T, \mathcal{M}) \leq k \Leftrightarrow \text{PF}(s, T, \mathcal{M}') \geq k'$. By definition of $x$, we have $\text{MFE}(s, T, \mathcal{M}) \leq k \Leftrightarrow \text{MFE}(s, T, \mathcal{M}) \leq x$. Therefore, it's equivalent to prove the following:

$$\text{MFE}(s, T, \mathcal{M}) \leq x \Leftrightarrow \text{PF}(s, T, \mathcal{M}') \geq k'.$$

The partition function, after magnification, is the following:

$$\mathrm{PF}(s,T,\mathcal{M}') = \sum_{g\in\mathcal{G}_s^{\mathcal{M}}} \#\mathrm{SSEL}(s,T,\mathcal{M},g)e^{-\beta\cdot g\cdot\ln(n!)/(\beta\delta)} = \sum_{g\in\mathcal{G}_s^{\mathcal{M}}} \#\mathrm{SSEL}(s,T,\mathcal{M},g)(n!)^{-g/\delta}$$

$\Rightarrow$ If $\mathrm{MFE}(s,T,\mathcal{M}) \leq x$, then a MFE secondary structure contributes to the magnified partition function, $\mathrm{PF}(s,T,\mathcal{M}')$, with a coefficient of $(n!)^{-\mathrm{MFE}(s,T,\mathcal{M})/\delta} \geq (n!)^{-x/\delta} = k'$, hence $\mathrm{PF}(s,T,\mathcal{M}') \geq k'$ as required.

$\Leftarrow$ Conversely, if $\mathrm{MFE}(s,T,\mathcal{M}) > x$, then by definition of $\delta$, $\mathrm{MFE}(s,T,\mathcal{M}) \geq x+\delta$.

$$\begin{aligned}
\mathrm{PF}(s,T,\mathcal{M}') &= \sum_{g\in\mathcal{G}_s^{\mathcal{M}}} \#\mathrm{SSEL}(s,T,\mathcal{M},g)(n!)^{-g/\delta} \\
&\leq \#\mathrm{SecStruct}(s)\cdot(n!)^{-\mathrm{MFE}(s,T,\mathcal{M})/\delta} \\
&< n!\cdot(n!)^{-\mathrm{MFE}(s,T,\mathcal{M})/\delta} && \text{As } \#\mathrm{SecStruct}(s) < n! \text{ by Lemma 21.} \\
&< (n!)^{-(\mathrm{MFE}(s,T,\mathcal{M})-\delta)/\delta} \\
&< (n!)^{-x/\delta} = k'
\end{aligned}$$

This proves the main claim that: $\mathrm{dMFE}(s,T,\mathcal{B},k) \Leftrightarrow \mathrm{dPF}(s,T,\mathcal{B}',k')$.

**Complexity analysis.**

- Step 1: is done in polynomial-time, since $|\mathcal{G}_s^{\mathcal{M}}| = \mathrm{poly}(n)$, Note 13.
- Step 2 (magnification step): is done in polynomial-time, since $\log(\frac{\log(n!)}{\beta\cdot\delta}) \leq \log(n^2) + \log(k_{\mathrm{B}}T) + \log(1/\delta) \leq \mathrm{poly}(n) + \mathrm{poly}(T)$, due to the logarithmic size of $1/\delta$, Note 13.
- Step 3: $k'$ can be computed in $\mathrm{poly}(n)$ time, and the size of $k'$ is $\log((n!)^{-x/\delta}) \leq (-x/\delta)\cdot n^2 = \mathrm{poly}(n)$ bits.

This shows the existence of a polynomial-time Turing reduction from dMFE to dPF. ◀

## 3.4 Polynomial-time Turing Reduction from PF to dPF

This next reduction also exploits our magnification trick, hence is slightly more involved than the previous one from MFE to dMFE. while for PF, the search space is of exponential size. A first approach would be to binary search for the PF value, using the oracle dPF. This approach runs in polynomial time if we are satisfied with linear precision of the PF value (but the PF can have arbitrary precision since it uses exponentials of e). However, we can search for the exact value by combining the simple binary search with our magnification trick, efficiently exploiting the dPF oracle to compute the exact PF contribution at each energy level. The proof of the following theorem is in Appendix B.2.

▶ **Theorem 24.** *There exists a polynomial-time Turing reduction from* PF *to* dPF.

## 3.5 Discussion for Section 3: reductions and the magnification trick

When we call a specific oracle using our *magnification trick*, we ask the oracle to function under the same energy model $\mathcal{M}$ but magnified (e.g. by factor $j$ in Line 3 of Algorithm 1, to give $\mathcal{M}_j$). We believe this kind of magnification of the energy model, combined with our reductions in Section 3, is a useful notion for finding polynomial time algorithms for thermodynamic problems (such as those in Figure 2), hence we propose a general definition:

▶ **Definition 25** (PF-polynomially magnifiable energy model)**.** An energy model $\mathcal{M}$ is **PF-polynomially magnifiable** if there is a polynomial time algorithm $\mathrm{PF}(\cdot,\cdot,\mathcal{M}_j)$ that computes PF under the $j$-magnified energy model: $\Delta G^{\mathcal{M}_j} = j\cdot\Delta G^{\mathcal{M}}$, for all $j \in \mathrm{poly}(n)$, and we call $\mathrm{PF}(\cdot,\cdot,\mathcal{M})$ a **magnification adaptable** algorithm under $\mathcal{M}$.

### 3.5.1 Positive (polynomial time) results

In the unpseudoknotted 1-strand, or even $\mathcal{O}(1)$-strand cases, with the BPM and BPS models, there is a polynomial time algorithm for PF. It is not hard to show that PF for BPM and BPS is magnification adaptable, hence BPM and BPS are PF-polynomially magnifiable energy models. Hence, from our reductions and that PF algorithm. we get polynomial time algorithms for free for the other four problems besides PF in Figure 2. The same holds for the NN model *without rotational symmetry* (i.e. single-stranded NN model, or the multi-stranded NN model that ignores rotational symmetry).

▶ **Corollary 26.** *When $\mathcal{M}$ is a PF-polynomially magnifiable model, all five problems in Figure 2 are in P.*

In the NN model with multiple strands and unpseudoknotted secondary structures, Dirks et al. [8] gave a polynomial time algorithm for PF, and Shalaby and Woods [26] gave one for MFE. One might ask if Dirks et al. [8], plus the reductions in Figure 2 are sufficient to yield a polynomial time algorithm for MFE, but this is not known since we don't know whether the Dirks et al. PF algorithm is magnification adaptable. The issue is in how Dirks et al. handle rotational symmetry, not by pure dynamic programming, but by computing a naive PF that: (1) Completely ignores rotational symmetry, and (2) Overcounts *indistinguishable* secondary structures (less symmetry, means more overcounting, see [8] for details). Then they use an algebraic argument to bring back rotational symmetry simultaneously with canceling out the overcounting effect, an argument that may not be preserved under our magnification trick. Hence we do not currently know whether the NN model in its full generality (including rotational symmetry) is PF-polynomially magnifiable.

### 3.5.2 Temperature independent energy models

If the energy model is not PF-polynomially magnifiable, but is temperature independent (Note 10), we can achieve our magnification trick simply by instead reducing (i.e. inverse magnification) the partition function temperature as follows (this assumes, as usual, that the partition function *is* temperature dependent):

- Assume we have a polynomial time algorithm $\mathrm{PF}(\cdot,\cdot,\mathcal{M})$ but want to compute $\mathrm{PF}(\cdot,\cdot,\mathcal{M}')$, where $\Delta G^{\mathcal{M}'} = \alpha \Delta G^{\mathcal{M}}$.
- Call $\mathrm{PF}(s, T', \mathcal{M})$, with $T' = T/\alpha$.

$$\begin{aligned}
\mathrm{PF}(s, T', \mathcal{M}) &= \sum_{S \in \Omega} \mathrm{e}^{-\Delta G^{\mathcal{M}}(S)/k_{\mathrm{B}}T'} \\
&= \sum_{S \in \Omega} \mathrm{e}^{-\alpha \Delta G^{\mathcal{M}}(S)/k_{\mathrm{B}}T} && \mathcal{M} \text{ is temperature independent.} \\
&= \mathrm{PF}(s, T, \mathcal{M}')
\end{aligned}$$

- Hence, $\mathrm{PF}(\cdot,\cdot,\mathcal{M})$ is magnification adaptable.

▶ Note 27 (NP-hardness results). As we mentioned before, the dMFE problem is NP-hard in: (1) The BPS model when we allow pseudoknots [19], and (2) The BPM model when we have unbounded number of strands [5] without pseudoknots. Then, #SSEL, PF, and dPF are NP-hard as well in these energy models using our reduction map, Figure 2. However, we provide **sharper** complexity results in the following sections.

## 4 #P-hardness of PF and #SSEL in BPS model

In this section, we prove #P-hardness for the problems PF$(\cdot,\cdot,\text{BPS})$ and #SSEL$(\cdot,\cdot,\text{BPS})$ (see Definition 9 for the BPS model). Our strategy is to construct a reduction chain from #3DM to #4-PARTITION, then from #4-PARTITION to #BPS, where #BPS is the counting problem of the number of secondary structures having exactly $K$ stacks.

First, we state the definitions of the three problems within the reduction chain, before showing that there exist weakly parsimonious reductions[5] within the chain.

▶ **Definition 28** (#3DM).
**Input:** Three finite sets $X$, $Y$, and $Z$ of the same size, and a subset $T \subseteq X \times Y \times Z$.
**Output:** The number of perfect 3-dimensional matchings $M$, where $M \subseteq T$ is a perfect 3-dimensional matching if any element in $X \cup Y \cup Z$ belongs to exactly one triple in $M$.

▶ **Definition 29** (#4-PARTITION).
**Input:** A set of $k$ elements $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$, a weight function $w : \mathcal{A} \to \mathbb{Z}^+$, and a bound $B \in \mathbb{Z}^+$ such that the weight of each element $w(a_i)$ is strictly between $B/5$ and $B/3$.
**Output:** The number of partitions of $\mathcal{A}$ into 4-tuples, such that the sum of the weights of the elements in each tuple is equal to $B$.

▶ **Definition 30** (#BPS).
**Input:** A nucleic acid strand $s$ and a number $K \in \mathbb{N}$.
**Output:** The number of secondary structures $S$ of $s$ that have $K$ base pair stackings.

We prove that #BPS is #P-complete through two consecutive weakly parsimonious reductions: from #3DM, to #4-PARTITION, and from #4-PARTITION to #BPS. The proof of the following lemma and theorem are in Appendix B.3.

▶ **Lemma 31.** *#4-PARTITION is #P-complete.*

▶ **Theorem 32.** *#BPS is #P-complete.*

▶ **Theorem 33.** #SSEL$(\cdot,\cdot,\text{BPS})$ *is #P-complete and* PF$(\cdot,\cdot,\text{BPS})$ *is #P-hard.*

**Proof.** #SSEL$(\cdot,\cdot,\text{BPS})$ belongs to #P and is equivalent to the problem #BPS. Hence, #SSEL$(\cdot,\cdot,\text{BPS})$ is #P-complete. The reduction map (Figure 2), specifically Theorem 22, implies that computing the partition function is #P-hard in the BPS model. ◀

▶ **Theorem 34.** *In the BPS model, there is no polynomial-time Turing reduction from* #SSEL *to dMFE, unless* $\#P \subseteq P^{NP}$.

**Proof.** dMFE is NP-complete in the BPS model according to Lyngsø [19], and #SSEL is #P-complete according to Theorem 33. Suppose that there exists a polynomial-time Turing reduction from #SSEL to dMFE (in the BPS model): $\#\text{SSEL} \in \text{P}^{\text{dMFE}}$.

As these problems are complete, $\#\text{SSEL} \in \text{P}^{\text{dMFE}} \Rightarrow \#\text{P} \subseteq \text{P}^{\text{NP}}$. ◀

▶ Remark 35. If $\#\text{P} \subseteq \text{P}^{\text{NP}}$, then $\text{P}^{\#\text{P}} \subseteq \text{P}^{\text{P}^{\text{NP}}} = \text{P}^{\text{NP}} = \Delta_2^{\text{P}} \subseteq \Sigma_2^{\text{P}}$. According to Toda's theorem [28], $\text{PH} \subseteq \text{P}^{\#\text{P}}$. Therefore, $\text{PH} \subseteq \Sigma_2^{\text{P}}$: the Polynomial Hierarchy would collapse at level 2.

[5] A reduction is weakly parsimonious if there is a suitably-computable relation between the number of solutions of the two problems; depending only on the initial problem instance [2].

## 5 #P-hardness of PF and #SSEL for unpseudoknotted secondary structures of an unbounded set of strands in the BPM model

In this section, we prove #P-hardness of the two restricted problems PF$(\cdot,\cdot,\text{BPM})$ and #SSEL$(\cdot,\cdot,\text{BPM})$, see Definition 8 (BPM model), in the scenario of unpseudoknotted structures of an unbounded set of strands. Specifically, we show that the problem #MULTI-PKF-SSP is #P-complete. This main result is proven using lemmas from Appendix B.4.

▶ **Definition 36** (#MULTI-PKF-SSP).
**Input:** $c$ nucleic acid strands and a positive integer $k$.
**Output:** The number of unpseudoknotted secondary structures containing exactly $k$ base pairs formed by the $c$ strands.

▶ **Theorem 37.** *#MULTI-PKF-SSP is #P-complete.*

**Proof.** To prove #P-completeness, we first consider the reduction provided by Condon, Hajiaghayi, and Thachuk for the NP-completeness of the associated decision problem [5]. Using the same notation, we show that this reduction is weakly parsimonious. Since this reduction contains many details, we do not recall all of them here.

We first note that the reduction in [5] is from a variation of 3-Dimensional Matching, 3DM(3), in which each element appears in at most 3 triples. Without loss of generality, this reduction can be extended from another version of 3DM, in which all triples are distinct.

Condon, Hajiaghayi, and Thachuk showed (in Lemma 12 of [5]) that any solution of 3DM, can be transformed into a secondary structure of the new instance $I'$ containing $P$ base pairs, where $P$ is the maximum number of possible base pairs ($P = \min(A,T) + \min(G,C)$). It is easy to see that this transformation from 3DM to MULTI-PKF-SSP is injective.

For the reverse transformation, we show an intermediate result in Lemma 43, ensuring the form of any secondary structure of the new instance $I'$ containing $P$ base pairs. This lemma is stated and proven in Appendix B.4. According to this lemma, the back transformation is clear: considering the perfect triples provides a solution for 3DM. Let two secondary structures leading to the same 3DM solution. Consequently, they have the same perfect and center-trim-deprived triples. Let us make a little dichotomy here:

- If we consider the strands as indistinguishable, then the two secondary structures are the same and the reduction is parsimonious.
- Else if we consider the strands as distinguishable, then the two secondary structures differ as some of their trim-complement and separator support strands are permuted. Since there are $2n$ separator strands, $2n$ separator-complement strands and $2m+n$ trim-complement strands, there are $(2n)!^2 \cdot (2m+n)!$ distinct secondary structures mapping to the same 3DM solution.

In both cases, there is a simple relation between the number of solutions of the two problems, and hence the reduction is weakly parsimonious.

#MULTI-PKF-SSP belongs to #P, since MULTI-PKF-SSP is in NP. Since #3DM is #P-complete even in the case of disctinct triples [3], #MULTI-PKF-SSP is also #P-complete. ◀

▶ **Theorem 38.** #SSEL$(\cdot,\cdot,\text{BPM})$ *is #P-complete and* PF$(\cdot,\cdot,\text{BPM})$ *is #P-hard, in the scenario of unpseudoknotted structures of an unbounded set of strands.*

**Proof.** It is clear that #SSEL$(\cdot,\cdot,\text{BPM})$ belongs to #P and is equivalent to the problem #MULTI-PKF-SSP in this scenario. Hence, #SSEL$(\cdot,\cdot,\text{BPM})$ is #P-complete.

The reduction map (Figure 2) and specifically Theorem 22 implies that computing the partition function is #P-hard in this scenario as well. ◀

## References


1 Tatsuya Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104(1):45–62, 2000. doi:10.1016/S0166-218X(00)00186-4.

2 Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

3 Jeffrey Bosboom, Erik D. Demaine, Martin L. Demaine, Adam Hesterberg, Roderick Kimball, and Justin Kopinsky. Path puzzles: Discrete tomography with a path constraint is hard. *Graphs and Combinatorics*, 36:251–267, 2020. doi:10.1007/s00373-019-02092-5.

4 Ho-Lin Chen, Anne Condon, and Hosna Jabbari. An $\mathcal{O}(n^5)$ algorithm for MFE prediction of kissing hairpins and 4-chains in nucleic acids. *Journal of Computational Biology*, 16(6):803–815, 2009. PMID: 19522664. doi:10.1089/cmb.2008.0219.

5 Anne Condon, Monir Hajiaghayi, and Chris Thachuk. Predicting minimum free energy structures of multi-stranded nucleic acid complexes is APX-hard. In Matthew R. Lakin and Petr Šulc, editors, *27th International Conference on DNA Computing and Molecular Programming (DNA 27)*, volume 205 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:21, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.DNA.27.9.

6 Jan Cupal, Ivo L. Hofacker, and Peter F. Stadler. Dynamic programming algorithm for the density of states of rna secondary structures. In Reinhard Hofstädt, Thomas Lengauer, Markus Löffler, and Dirk Schomburg, editors, *Computer Science and Biology'96 (German Conference on Bioinformatics)*, pages 184–186, Leipzig, Germany, 1996. University of Leipzig.

7 Erik D Demaine, Timothy Gomez, Elise Grizzell, Markus Hecher, Jayson Lynch, Robert Schweller, Ahmed Shalaby, and Damien Woods. Domain-based nucleic-acid minimum free energy: Algorithmic hardness and parameterized bounds. In *30th International Conference on DNA Computing and Molecular Programming (DNA 30)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.DNA.30.2.

8 Robert M Dirks, Justin S Bois, Joseph M Schaeffer, Erik Winfree, and Niles A Pierce. Thermodynamic analysis of interacting nucleic acid strands. *SIAM review*, 49(1):65–88, 2007. doi:10.1137/060651100.

9 Robert M Dirks and Niles A Pierce. A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of computational chemistry*, 24(13):1664–1677, 2003. doi:10.1002/JCC.10296.

10 Robert M Dirks and Niles A Pierce. An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots. *Journal of computational chemistry*, 25(10):1295–1304, 2004. doi:10.1002/JCC.20057.

11 Gwendal Ducloz, Ahmed Shalaby, and Damien Woods. Algorithmic hardness of the partition function for nucleic acid strands, 2025. Extended arXiv version of this paper: arXiv:2506.19756.

12 Mark E Fornace, Nicholas J Porubsky, and Niles A Pierce. A unified dynamic programming framework for the analysis of interacting nucleic acid strands: enhanced models, scalability, and speed. *ACS Synthetic Biology*, 9(10):2665–2678, 2020.

13 Michael R Garey and David S Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1979.

14 Leslie Ann Goldberg, Martin Grohe, Mark Jerrum, and Marc Thurley. A complexity dichotomy for partition functions with mixed signs. *SIAM Journal on Computing*, 39(7):3336–3402, 2010. doi:10.1137/090757496.

15 Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

16 Hosna Jabbari, Ian Wark, Carlo Montemagno, and Sebastian Will. Knotty: efficient and accurate prediction of complex RNA pseudoknot structures. *Bioinformatics*, 34(22):3849–3856, 2018. doi:10.1093/BIOINFORMATICS/BTY420.

17 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*, 6(1):26, 2011. doi:10.1186/1748-7188-6-26.

18 Feng Lou and Peter Clote. Thermodynamics of RNA structures by Wang–Landau sampling. *Bioinformatics*, 26(12):i278–i286, 2010.

19 Rune B Lyngsø. Complexity of pseudoknot prediction in simple models. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming*, pages 919–931, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. doi:10.1007/978-3-540-27836-8_77.

20 Rune B Lyngsø and Christian NS Pedersen. RNA pseudoknot prediction in energy-based models. *Journal of computational biology*, 7(3-4):409–427, 2000. doi:10.1089/106652700750050862.

21 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.

22 Ruth Nussinov and Ann B Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proceedings of the National Academy of Sciences*, 77(11):6309–6313, 1980.

23 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.

24 Elena Rivas and Sean R Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of molecular biology*, 285(5):2053–2068, 1999.

25 John SantaLucia Jr and Donald Hicks. The thermodynamics of DNA structural motifs. *Annual Review of Biophysics*, 33:415–440, 2004. doi:10.1146/annurev.biophys.32.110601.141800.

26 Ahmed Shalaby and Damien Woods. An efficient algorithm to compute the minimum free energy of interacting nucleic acid strands. In *ICALP: The 52nd International Colloquium on Automata, Languages and Programming*, Leibniz International Proceedings in Informatics (LIPIcs), 2025. To appear. arXiv version: arXiv:2407.09676. doi:10.48550/arXiv.2407.09676.

27 Ignacio Tinoco, Olke C Uhlenbeck, and Mark D Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230(5293):362–367, 1971.

28 Seinosuke Toda. On the computational power of PP and $\oplus$P. In *FOCS: 54th Annual Symposium on Foundations of Computer Science*, pages 514–519. IEEE Computer Society, 1989. doi:10.1109/SFCS.1989.63527.

29 Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979. doi:10.1137/0208032.

30 John N. Zadeh, Christopher D. Steenberg, Justin S. Bois, Blake R. Wolfe, Matthew B. Pierce, Abbas R. Khan, Raymond M. Dirks, and Niles A. Pierce. NUPACK: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–173, 2011. doi:10.1002/JCC.21596.

31 Michael Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research*, 31(13):3406–3415, 2003. doi:10.1093/NAR/GKG595.

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

33 Michael Zuker and Patrick Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic acids research*, 9(1):133–148, 1981. doi:10.1093/NAR/9.1.133.

# A Appendix: Candidate energy levels for three free energy models

We state two lemmas, the proofs of which are in the full version of this work [11]. The first states that for an instance of any of the three energy models, BPM, BPS and NN, there is a set of candidate energy levels of polynomial size and polynomial time computable in number of bases $n$. For the NN model, the second lemma implies two more efficient and refined algorithms that better approximate the energy levels occupied by secondary structures.

▶ **Lemma 39.** *Given a multiset of strands $s$ and temperature $T$, for each of the BPS, BPM and NN models, there is a set of candidate energy levels $\mathcal{G}_s^{\mathcal{M}}$ of polynomial size and computable in polynomial time in the number of bases $n$. More precisely, the size $|\mathcal{G}_s^{\mathcal{M}}|$ is linear for BPS and BPM, and $O(n^{O(1)})$ for NN.*

▶ **Lemma 40.** *Given a multiset of strands $s$ and temperature $T$ for the NN model, there is an $\mathcal{O}(n^4|\mathcal{G}'|^2)$ algorithm to compute a set of candidate energy levels $\mathcal{G}_s^{NN}$, where $|\mathcal{G}'| = n^{O(1)}$ is defined in the proof. The computed set is precisely the occupied energy levels (that have at least one structure) if we ignore rotational symmetry. A second algorithm allows for rotational symmetry, but gives a (typically strict) superset of the occupied energy levels.*

# B Appendix: Omitted proofs

## B.1 Proof of Lemma 21, omitted from Section 3

▶ **Lemma 21.** *For any strand $s$ of size $n > 2$, $\#\text{SecStruct}(s) < n!$.*

**Proof.** By enumerating all secondary structures based on the number of base pairs formed in each. Any secondary structure $S$ has $k \leq \lfloor n/2 \rfloor$ base pairs, these $k$ base pairs are formed between $2k$ different bases. The first base, among the $2k$ different bases, has $2k-1$ bases to form a base pair with. The next "unpaired" base has $(2k-3)$ bases to form a base pair with. At the end, the number of secondary structures which have exactly $k$ base pairs is upper bounded by $\binom{n}{2k} \cdot (2k-1) \cdot (2k-3) \cdots 3 \cdot 1$. Leading to

$$\#\text{SecStruct}(s) \leq \sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{2k} \cdot (2k-1) \cdot (2k-3) \cdots \tag{2}$$

$$\leq \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{n!}{(2k)!\ (n-2k)!} \frac{1 \cdot 2 \cdot 3 \cdots 2k}{2 \cdot 4 \cdot 6 \cdots 2k} \tag{3}$$

$$\leq \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{n!}{(2k)!\ (n-2k)!} \frac{(2k)!}{k!\ 2^k} \tag{4}$$

$$\leq n! \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{1}{(n-2k)!\ k!\ 2^k} \tag{5}$$

$$\leq n! \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{1}{\lfloor n/3 \rfloor!\ 2^k} \qquad \text{as } \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{1}{2^k} < 2 \tag{6}$$

$$< \frac{2n!}{\lfloor n/3 \rfloor!} \leq n! \qquad \text{for } n \geq 6 \tag{7}$$

Note that, $\#\mathrm{SecStruct}(s) = n!$ for $n = 1$ or $n = 2$, which are useless cases for any system. To extend the inequality to all $n > 2$, we only need to check if it is verified for $3 \leq n \leq 5$. By computing the right-hand side of Equation (2), the associated numbers of possible secondary structures are at most $4, 10,$ and $26$ which are upper bounded by its corresponding $n!$. Therefore, $\#\mathrm{SecStruct}(s) < n!$ for all $n > 2$. ◀

## B.2 Proof of Theorem 24, omitted from Section 3

▶ **Theorem 24.** *There exists a polynomial-time Turing reduction from* PF *to* dPF*.*

**Proof.** Let $(s, T, \mathcal{M})$ be the input of the PF problem, and consider Algorithm 2 and notations used in Theorem 23. Since $\#\mathrm{SSEL}(s, T, \mathcal{M}, g) < n!$ for all $g \in \mathcal{G}_s^{\mathcal{M}}$ by Lemma 21, the partition function $\mathrm{PF}(s, T, \mathcal{M}') = \sum_{g \in \mathcal{G}_s^{\mathcal{M}}} \#\mathrm{SSEL}(s, T, \mathcal{M}, g)(n!)^{-g/\delta}$ can be seen as a number written in base $n!$. Determining this number in base $n!$ is equivalent to determining all coefficients of the form $\#\mathrm{SSEL}(s, T, \mathcal{M}, g)$. To achieve this, we do the following:

- We perform a binary search for each coefficient $\#\mathrm{SSEL}(s, T, \mathcal{M}, g)$ in decreasing order over $\mathcal{G}_s^{\mathcal{M}}$ (from the most to the less favourable).
- First, guess $\#\mathrm{SSEL}(s, T, \mathcal{M}, g_{|\mathcal{G}_s^{\mathcal{M}}|})$, by calling $\mathrm{dPF}(s, T, \mathcal{M}', c_1(n!)^{-g_{|\mathcal{G}_s^{\mathcal{M}}|}/\delta})$, repeatedly, using binary search with search variable $c_1$ over the range $0 \leq c_1 \leq n!$.
- Inductively, after computing $\#\mathrm{SSEL}(s, T, \mathcal{M}, g_i)$, find $\#\mathrm{SSEL}(s, T, \mathcal{M}, g_{i-1})$ by calling dPF oracle with input $(s, T, \mathcal{M}', \sum_{j=1}^{i} \#\mathrm{SSEL}(s, T, \mathcal{M}, g_j)(n!)^{-g_j/\delta} + c_{i-1}(n!)^{-g_{i-1}/\delta})$, where $c_{i-1}$ is the search variable: $0 \leq c_{i-1} \leq n!$.
- At the end, we know all of the $\#\mathrm{SSEL}(s, T, \mathcal{M}, g_i)$ values, we then combine them to directly compute the partition function $\mathrm{PF}(s, T, \mathcal{M}) = \sum_{i=1}^{N} \#\mathrm{SSEL}(s, T, \mathcal{M}, g_i)e^{-g_i/(k_{\mathrm{B}}T)}$.

By using binary search, the algorithm requires only $|\mathcal{G}_s^{\mathcal{M}}| \log(n!)$ calls to dPF oracles, which is $\mathrm{poly}(n)$, Note 13. Hence, it is a polynomial-time Turing reduction from PF to dPF. ◀

## B.3 Proof of Lemma 31, omitted from Section 4

▶ **Lemma 31.** *#4-PARTITION is #P-complete.*

For the sake of completeness, we first include a quick overview of the 4-PARTITION NP-completeness from Garey and Johnson (1979) [13], as we heavily depend on it in our reduction. After that we give the proof of Lemma 31.

**Proof overview of the NP-completeness of 4-PARTITION.**
Given an instance of 3DM with three disjoint sets $X$, $Y$, $Z$, with $|X| = |Y| = |Z|$, and a set $T \subseteq X \times Y \times Z$, the reduction constructs an instance $(\mathcal{A}, w, B)$ of 4-PARTITION as follows:

- For each element $x \in X$, we introduce $N(x)$ elements $(x[i])_{1 \leq i \leq N(x)}$ in $\mathcal{A}$, where $N(x)$ is the number of times $x$ appears in $T$. These elements have two possible weights, depending on whether $i = 1$ or not. The element $x[1]$ is called *actual* while the other ones are called *dummy*. Same happens to every $y \in Y$, and $z \in Z$.
- For each triple $(x, y, z) \in T$, an element $u_{xyz}$, is introduced in $\mathcal{A}$, whose weight depends on $x$, $y$ and $z$.
- The bound $B$ and the weight function $w$ are constructed such that the following equivalence hold: A 4-tuples has weight equaling $B$ iff it is of the form $(u_{xyz}, x[i], y[j], z[k])$, with $i, j, k$ being all equal to 1, or all greater than 1. In that case, we call the 4-tuple either *dummy* or *actual*.

Given a 3D-matching $M$, we can construct the following 4-partition $P$. For each triple $(x, y, z) \in M$ , we add to $P$ the *actual* 4-tuple $(u_{xyz}, x[1], y[1], z[1])$. For each triple $(x, y, z) \in T \setminus M$, we add to $P$ the *dummy* 4-tuple $(u_{xyz}, x[i], y[j], z[k])$, with $i, j$,and $k$ all greater than 1. It is possible to construct these triples since there are enough *dummy* elements. $P$ is a solution of 4-PARTITION since it is a partition (all elements are part of a 4-tuple) and since the weight of each 4-tuple equals $B$ by construction.

Reciprocally, given a 4-partition $P$ of $\mathcal{A}$, we construct a 3D-matching $M$. All 4-tuples of $P$ have weight equaling $B$ and are of the expected form. We introduce in $M$ the triples $(x, y, z) \in T$ for the ones $u_{xyz}$ is part of an *actual* 4-tuple in $P$. $M$ is a matching since there is exactly one *actual* element $a[i]$ per element $a \in X \cup Y \cup Z$. Moreover, $M$ is perfect since every *actual* element belongs to an *actual* 4-tuple. ◀

**Proof.** Now, we prove that in the mentioned reduction from 3DM to 4-PARTITION [13], the number of solutions (4-tuples) equals the number of perfect matchings in the initial instance of 3DM multiplied by a coefficient that depends only on that initial instance.

Let $M$ and $M'$ two distinct perfect 3D-matchings. There exists $(x, y, z) \in M \setminus M'$. Therefore, $(u_{xyz}, x[1], y[1], z[1])$ belongs to a 4-partition that corresponds to $M$, while the 4-tuple $(u_{xyz}, x[i], y[j], z[k])$ in a 4-partition corresponding to $M'$, must have $i$, $j$, and $k$ being greater than 1. Therefore, the two 4-partitions corresponding to $M$ and $M'$ are distinct.

Reciprocally, let $P$ and $P'$ are two 4-partitions corresponds to the same 3D-matching. Each tuple of $P$ and $P'$ is of the form $(u_{xyz}, x[i], y[j], z[k])$ where $(x, y, z) \in T$ and $i, j, k$ are equal to 1 or greater than 1. The Garey and Johnson's proof states that these two partitions have the same collections of *actual* 4-tuples. Therefore, they differ in their *dummy* 4-tuples (with $i, j, k$ greater than 1), which means that some *dummy* elements happens in different order.

For each element $a \in X \cup Y \cup Z$, there are $N(a) - 1$ *dummy* elements. Therefore, there are $\alpha = \prod_{a \in X \cup Y \cup Z}(N(a) - 1)!$ different ways to arrange these elements in dummy collections. We can conclude that there are $\alpha$ distinct 4-partitions corresponding to the same 3D-matching.

This implies that the number of solutions of 4-PARTITION equals the number of perfect matchings in the initial instance of 3DM multiplied by multiplied by $\alpha$, hence this reduction is weakly parsimonious.

It is straighforward that #4-PARTITION belongs to #P as 4-PARTITION belongs to NP. Since #3DM is #P-hard according to Bosboom et al. [3], then #4-PARTITION is #P-complete. ◀

▶ **Theorem 32.** *#BPS is #P-complete.*

**Proof.** In 2004, Lyngsø [19] proved that the decision problem BPS is NP-complete, with a reduction from BIN-PACKING. We adapt his reduction from 4-PARTITION to BPS. Let $(\mathcal{A} = \{a_1, \cdots, a_k\}, w, B)$ be an instance of 4-PARTITION. Let us first note that in any instance of 4-PARTITION, $k$ is a multiple of 4, and $Bk/4 = \sum_{i=1}^{k} w(a_i)$, otherwise, it is straightforward to see that this instance has no solution. We proceed exactly as Lyngsø, constructing the following DNA strand $s$:

$$s = C^{w(a_1)}AC^{w(a_2)}A \cdots AC^{w(a_k)}AAA\underbrace{G^BAG^BA \cdots AG^B}_{k/4 \text{ substrings of } G's}$$

and setting the target $K = \sum_{i=1}^{k} w(a_i) - k$.

As A bases can only form base pairs with T bases, all base pairs in a secondary structure of s will be C-G base pairs. If a substring $C^{w(a_i)}$ binds with at least two disctinct substrings $G^B$, then it accounts for at most $w(a_i) - 2$ in the BPS score. Since the other substrings $C^{w(a_j)}$ account for at most $w(a_j) - 1$, then $\text{BPS}(S) \leq K - 1$. Hence, we can find a structure S with $\text{BPS}(S) = K$ iff we can partition the $k$ substrings $C^{w(a_i)}$ into $k/4$ groups that can each be fully base paired using one substring $G^B$; i.e. the total length of the substrings of C's in any group can be at most $B$.

It means that $\text{BPS}(S) = K$ iff we can partition the $k$ elements $a_i$ into $k/4$ groups that each has total weight $\leq B$. Since $B = \frac{4}{k}\sum_{i=1}^{k} w(a_i)$ and since the weight of each element in $\mathcal{A}$ is strictly between $B/5$ and $B/3$, each group has a total weight of exactly $B$ and contains exactly 4 elements. Therefore, $\text{BPS}(S) = K$ iff 4-PARTITION problem has a solution.

Consider two distinct 4-partitions $P_1$ and $P_2$. Let $(a_i, a_j, a_l, a_m)$ belong to $P_1$ but not to $P_2$. The secondary structure corresponding to $P_1$ has the group ($C^{w(a_i)}$, $C^{w(a_j)}$, $C^{w(a_l)}$, $C^{w(a_m)}$) fully bounded with a substring $G^B$. The secondary structure corresponding to $P_2$ does not have these bindings, as $(a_i, a_j, a_l, a_m)$ does not belong to $P_2$. Therefore, all distinct 4-partitions map to distinct secondary structures.

Conversely, let $S_1$ and $S_2$ two secondary structures of $s$, having $K$ base pair stackings and corresponding to the same 4-partition. Consequently, the groups of 4 elements bounded with the substrings $G^B$ are the same in $S_1$ and $S_2$, up to permutations. Specifically, it means that $(k/4)!\ (4!)^{k/4}$ secondary structures of $s$ having $K$ BPS map to the same 4-PARTITION solution. The coefficient $(k/4)!$ accounts for the permutation over the substrings $G^B$, while the coefficient 4! accounts for the permutation between the substrings $C^{w(a_i)}$ within a same group (i.e. paired to a same substring $G^B$).

Therefore, the number of solutions is multiplied by $(k/4)!\ (4!)^{k/4}$ and this reduction is weakly parsimonious. Moreover, it is straightforward to see that #BPS is in #P, as BPS is in NP. Since #4-PARTITION is #P-complete by Lemma 31, #BPS is #P-complete also. ◀

## B.4 Omitted lemmas and proofs from Section 5

▶ **Lemma 41.** *Let $S$ be an optimal secondary structure of $I'$ with $P$ base pairs. In $S$, if a trim-complement strand is bound with a center-trim, it cannot also be bound with another trim.*

**Proof.** By absurd, suppose that $S$ has a trim-complement strand which is bound with a center-trim but also with another trim. Since all C's are paired in $S$, two adjacent bases of this trim-complement strand are paired with bases belonging to distinct domains $Trim_1$ and $Trim_2$. Since one of these two domains is a center-trim, they are non-adjacent and are separated by the non-empty sequence $u$. Since there is no pseudoknot, bases of $u$ can only be paired with each other. Without loss of generality, we can consider that $Trim_2$ is a middle-trim. Therefore, $u$ ends with a flank $x\ Sep\ y\ Sep\ z$. Since all A and T bases are paired in $S$ and since there is no pseudoknot, two adjacent bases of this flank must be paired with two adjacent bases of $u$. Therefore, $u$ must contain the complementary flank $\overline{z}\ \overline{Sep}\ \overline{y}\ \overline{Sep}\ \overline{x}$, which is absurd since all triples are different by hypothesis and since all xyz-support strands and their complement are distinct by construction. ◀

▶ **Lemma 42.** *Let $S$ be an optimal secondary structure of $I'$ with $P$ base pairs. $S$ has exactly $n$ perfectly bound center-trim domains.*

**Proof.** Let us first show that $S$ has at least $n$ perfectly bound center-trim domains. Since there are $2m+n$ trim-complement strands and every C's are paired in $S$, at least $nE$ bases belonging to center-trim domains are paired. According to the previous lemma, if one base of a center-trim is paired, then the entire domain is paired. Therefore, there are at least $n$ perfectly bound center-trim domains.

Let us now suppose that there are $n+i$ bound center-trim domains, with $1 \leq i \leq m-n$. They are perfectly bound according to the previous lemma. We call a flank isolated if its neighboring trim-domains are both bound with trim-complementary strands. Let us count the number of isolated flanks. There are $m-n-i$ center-trim deprived triples. All of them are bound to at most $2(m-n-i)E$ trim-complement bases. Therefore, there are at least $[2m+n-2(m-n-i)-(n+i)] \cdot E = (2n+i)E$ end-trim bases that participate in creating isolated flanks with one of the $n+i$ bound center-trims. Therefore, there are at least $(2n+i)$ isolated flanks.

For each isolated flank, consider the set of strands to which it is bound. This set is not bound to the rest of the template domain since there is no pseudoknot. According to Lemma 10 of Condon et al. [5], the total number of support strands is $10n$. Since there are at least $2n+i$ isolated flanks, at least one of them is bound to less than 5 support strands. We can apply Lemma 17 of Condon et al. [5] to this flank, since the ACT-unpairedness is 0 in $S$. Therefore, this flank must be bound to exactly 5 support strands. Recursively, the lemma assumptions still hold, and one can continue applying it until one has 0 available support strand but still $i$ isolated flanks respecting the assumptions, which is absurd. Therefore, we know that at most $n$ center-trim domains are perfectly bound.

Finally, there are exactly $n$ perfectly bound center-trim domains. ◀

▶ **Lemma 43.** *Each optimal secondary structure of instance $I'$ having $P$ base pairs is of the following form: $n$ perfect triples, $m-n$ triples whose both end-trims are bound to trim-complement strands, and whose 5' and 3' flanks are paired together.*

**Proof.** Let $S$ be an optimal secondary structure of $I'$ with $P$ base pairs.

According to Lemma 42, $S$ has exactly $n$ fully connected center-trim domains. The remaining trim-complement strands must necessarily be fully bound with trim domains, as there is no alternative configuration for them.

Following the same reasoning as in the previous proof, we recursively apply Lemma 17 from Condon et al. to the isolated flanks. Consequently, all support strands are bound. Since there are no unpaired A or T bases, each xyz-domain is paired with its complement.

To conclude, we note that the 5' and 3' flanks of trim-deprived triples must be bound together. ◀