Garcia-Martin *et al. BMC Bioinformatics* (2016) 17:424
DOI 10.1186/s12859-016-1280-6

BMC Bioinformatics

SOFTWARE Open Access

# RNAdualPF: software to compute the dual partition function with sample applications in molecular evolution theory

Juan Antonio Garcia-Martin[1,3], Amir H. Bayegan[1], Ivan Dotu[2] and Peter Clote[1*]

## Abstract

**Background:** RNA inverse folding is the problem of finding one or more sequences that fold into a user-specified target structure $s_0$, i.e. whose minimum free energy secondary structure is identical to the target $s_0$. Here we consider the ensemble of all RNA sequences that have low free energy with respect to a given target $s_0$.

**Results:** We introduce the program `RNAdualPF`, which computes the *dual partition function* $Z^*$, defined as the sum of Boltzmann factors $\exp(-E(\mathbf{a}, s_0)/RT)$ of all RNA nucleotide sequences **a** compatible with target structure $s_0$. Using `RNAdualPF`, we efficiently sample RNA sequences that approximately fold into $s_0$, where additionally the user can specify IUPAC sequence constraints at certain positions, and whether to include dangles (energy terms for stacked, single-stranded nucleotides). Moreover, since we also compute the *dual partition function* $Z^*(k)$ over all sequences having GC-content $k$, the user can require that all sampled sequences have a precise, specified GC-content.
Using $Z^*$, we compute the *dual expected energy* $\langle E^* \rangle$, and use it to show that natural RNAs from the `Rfam` 12.0 database have *higher* minimum free energy than expected, thus suggesting that functional RNAs are under evolutionary pressure to be only marginally thermodynamically stable.
We show that *C. elegans* precursor microRNA (pre-miRNA) is significantly *non-robust* with respect to mutations, by comparing the robustness of each wild type pre-miRNA sequence with 2000 [resp. 500] sequences of the same GC-content generated by `RNAdualPF`, which approximately [resp. exactly] fold into the wild type target structure.
We confirm and strengthen earlier findings that precursor microRNAs and bacterial small noncoding RNAs display plasticity, a measure of structural diversity.

**Conclusion:** We describe `RNAdualPF`, which rapidly computes the *dual partition function* $Z^*$ and samples sequences having low energy with respect to a target structure, allowing sequence constraints and specified GC-content. Using different inverse folding software, another group had earlier shown that pre-miRNA is mutationally robust, even controlling for compositional bias. Our opposite conclusion suggests a cautionary note that computationally based insights into molecular evolution may heavily depend on the software used.
C/C++-software for `RNAdualPF` is available at http://bioinformatics.bc.edu/clotelab/RNAdualPF.

**Keywords:** RNA secondary structure, Partition function, Boltzmann ensemble, Robustness

*Correspondence: clote@bc.edu
[1] Biology Department, Boston College, 140 Commonwealth Avenue, 02467 Chestnut Hill, MA, USA
Full list of author information is available at the end of the article

## Background

In [1], Borenstein and Ruppin define *neutrality* of an RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ by $\eta(\mathbf{a}) = 1 - \frac{\langle d \rangle}{n}$, where in this section $\langle d \rangle$ denotes the average, taken over all $3n$ single-point mutants of **a**, of the base pair distance $d_{\mathrm{BP}}$ between the minimum free energy (MFE) structure $s_0$ of **a** and the MFE structures of single-point mutants of **a**. An RNA sequence **a** is then defined to be *robust* if $\eta(\mathbf{a})$ is greater than the average neutrality of 1000 control sequences generated by the program `RNAinverse` [2], which fold into the same target structure $s_0$. The main finding of [1] is that precursor microRNAs (pre-miRNA) exhibit a significantly higher level of mutational robustness than random RNA sequences having the same structure. To control for sequence composition bias in their computational study, the authors selected sequences from the output of `RNAinverse`, whose dinucleotide composition was similar to that of wild type pre-miRNA (Jensen-Shannon divergence less than 0.01). Since the filtering step required enormous run time and computational resources, the authors restricted their attention to a small set of 211 microRNAs, generating only 100 control sequences per microRNA. Borenstein and Ruppin conclude that robustness of precursor microRNAs is not the byproduct of a base composition bias or of thermodynamic stability.

Subsequently Rodrigo et al. [3] undertook a similar analysis for bacterial small RNAs, also using the program `RNAinverse`, albeit using somewhat different definitions – precise definitions are given in “Formal definitions of robustness” section. The main finding of [3] was that bacterial sncRNAs are not significantly robust when compared with 1000 sequences having the same structure, as computed by `RNAinverse`; however, bacterial sncRNAs tend to be significantly *plastic*, in the sense that the ensemble of low energy structures is structurally diverse. Unlike the case of precursor microRNAs [1], Rodrigo et al. did not control for sequence compositional bias.

This raises the question of whether the control sequences analyzed in [1, 3] are *representative* or to what extent features shared by sequences output by the program `RNAinverse` are artifacts of the program used. Indeed, the number of RNA sequences that fold into a given target structure can be astronomically large. Over a few weeks, before we elected to terminate the execution, our state-of-the-art inverse folding software `RNAiFold` [4] generated 273,926,421 many 52-nt sequences that fold exactly into the MFE secondary structure $s_0$ of HIV-1 ribosomal frameshift stimulating signal from the Gag-Pol overlap region AF033819.3/1631-1682, and which additionally code 17-mer peptides in the Gag and Pol reading frames having amino acids that appear in Gag/Pol peptides found in the Los Alamos HIV-1 database [5]. The number of 52 nt RNA sequences that fold into target $s_0$ without additionally imposing the constraint of coding particular peptides in overlapping Gag/Pol reading frames is certain to dwarf the previous number. Moreover, the number of sequences that fold into the MFE structure of an animal precursor microRNA (length 68 to 91 nt [6]) or into the MFE structure of bacterial sncRNA (length 53-436 nt [3]) is certain to be even more daunting.

Different inverse folding algorithms have adopted different strategies to generate sequences that fold into a user-specified target secondary structure $s_0$. For instance, `RNAinverse` [2, 7] performs an *adaptive walk*, in one step of which a nucleotide in the current sequence is mutated and subsequently accepted if the base pair distance between the minimum free energy (MFE) structure of the mutated sequence and the target structure $s_0$ is reduced. `NUPACK Design` [8] selects a candidate mutation position with probability proportional to its contribution to the *ensemble defect* (Boltzmann-weighted Hamming distance to the vector representation of $s_0$, where $s_0[i] = j$ indicates $(i,j) \in s_0$ and $s_0[i] = i$ indicates $i$ is unpaired in $s_0$). `RNAiFold CP-design` [4, 9] uses constraint programming to systematically explore the search tree of all inverse folding solutions in an order determined by certain heuristics. Accordingly, one cannot claim that the collection of sequences generated by any particular inverse folding algorithm is *representative* of the astronomically large space of all inverse folding solutions – indeed, each inverse folding algorithm has an inherent but *unknown* bias.

In this paper, we describe the algorithm `RNAdualPF`, which generates sequences which have low free energy with respect to a user-specified target structure $s_0$ – i.e. the inherent bias of `RNAdualPF` is known, unlike the situation of other inverse folding algorithms. We show that `RNAdualPF` is extremely fast software for generating sequences that *approximately* fold into $s_0$; moreover, in a postprocessing step, one can filter the output of `RNAdualPF` to select sequences that exactly fold into $s_0$. `RNAdualPF` additionally allows the user to specify IUPAC codes to constrain certain nucleotide positions as well as to control the GC-content of all generated sequences. Sampling is performed in a manner distinct but somewhat analogous to that by which `Sfold` [10] and `RNAsubopt -p` [2] sample representative secondary structures from the Boltzmann ensemble of all structures of a given sequence. Using `RNAdualPF`, we perform a pilot study that is similar, though not identical, to that of [1, 3] for two classes of RNA: 250 *C. elegans* precursor microRNA from miRBase [11] and the bacterial small noncoding RNAs previously analyzed in [3].

Finally, it should be noted that, although `RNAdualPF` was developed entirely independently of the work of Reinharz et al. [12], one can view our C-program as an extension of Python program `IncaRNAtion` [12] to the full Turner energy model, where additionally GC-content is rigorously handled. This point will be discussed further in the Conclusion.

### Formal definitions of robustness

Let $\mathbf{a} = a_1, \ldots, a_n$ denote an arbitrary RNA sequence, where $a_i \in \mathcal{N} = \{\mathrm{A, U, G, C}\}$, a secondary structure $s$ of $\mathbf{a}$ is a set of base pairs $(i,j)$ satisfying the following conditions: (1) If $(i,j) \in s$ then $a_i, a_j$ constitute a Watson-Crick or GU wobble pair, i.e. $ij \in \mathcal{B}$ which is the set {AU,UA,GC,CG,GU,UG}. (2) If $(i,j) \in s$ then $i + \theta < j$, where $\theta = 3$ (a minimum assumed for steric hindrance). (3) If $(i,j) \in s$ and $(k,\ell) \in s$, then either $i < k < \ell < j$ or $k < i < j < \ell$ or $i < j < k < \ell$ or $k < \ell < i < j$. The collection of all secondary structures of the RNA sequence $\mathbf{a}$ is denoted $\mathbb{SS}(\mathbf{a})$, and the free energy [13] of $s$ is denoted by $E(\mathbf{a}, s)$, or simply by $E(s)$ provided that the sequence $\mathbf{a}$ is clear from context. The *Boltzmann probability* $p(s) = p_{\mathbf{a}}(s)$ for structure $s$ of $\mathbf{a}$ is defined by $\exp(-E(\mathbf{a},s)/RT)/Z$, where the partition function $Z = Z(\mathbf{a}) = \sum_{s\in\mathbb{SS}(\mathbf{a})} \exp(-E(\mathbf{a},s)/RT)$. Given two secondary structures $s, t$ of $\mathbf{a}$, the *base pair distance* $d_{\mathrm{BP}}(s,t)$ between $s$ and $t$ is defined to be the size of the symmetric difference of $s, t$, i.e. $|s - t| + |t - s|$.

In [3], Rodrigo et al. define *intrinsic distance*

$$d_0(\mathbf{a}) = \sum_{s,t} p(s) \cdot p(t) \cdot d_{\mathrm{BP}}(s,t) \tag{1}$$

i.e. intrinsic distance is another name for *ensemble diversity* earlier defined in [14], and computed by `Vienna RNA Package` [2]. *Plasticity* is defined in [3] to be *normalized ensemble diversity*; i.e.

$$P(\mathbf{a}) = \frac{d_0(\mathbf{a})}{n/2} \tag{2}$$

obtained by dividing ensemble diversity by (essentially) the maximum possible number $n/2$ of base pairs in a structure of $\mathbf{a}$. Given two RNA sequences $\mathbf{a} = a_1, \ldots, a_n$ and $\mathbf{b} = b_1, \ldots, b_n$ of the same length $n$, Rodrigo et al. define $d_1(\mathbf{a}, \mathbf{b})$ to be the expected base pair distance between structures of $\mathbf{a}$ and structures of $\mathbf{b}$ minus the ensemble diversity of $\mathbf{a}$, i.e.

$$d_1(\mathbf{a},\mathbf{b}) = \sum_{s\in\mathbb{SS}(\mathbf{a})} \sum_{t\in\mathbb{SS}(\mathbf{b})} p_{\mathbf{a}}(s) \cdot p_{\mathbf{b}}(t) \cdot d_{\mathrm{BP}}(s,t) - d_0(\mathbf{a}) \tag{3}$$

Since $d_1$ is not symmetric, this measure is not a metric. In contrast, *ensemble distance* as described in [14] is a valid metric, defined by the following:

$$\begin{aligned} D_{\mathrm{V}}(\mathbf{a},\mathbf{b}) &= \sqrt{\sum_{s\in\mathbb{SS}(\mathbf{a})} \sum_{t\in\mathbb{SS}(\mathbf{b})} p_{\mathbf{a}}(s) \cdot p_{\mathbf{b}}(t) \cdot d_{\mathrm{BP}}(s,t) - \frac{d_0(\mathbf{a}) + d_0(\mathbf{b})}{2}} \\ &= \sqrt{\sum_{i<j} (p_{i,j}(a) - p_{i,j}(b))^2} \end{aligned} \tag{4}$$

In [3], Rodrigo et al. define the *mutational robustness*

$$R_m(\mathbf{a}) = \qquad 1 - \frac{\langle d_1(\mathbf{a}, \mathbf{a'})\rangle}{n/2} \tag{5}$$

where $\langle d_1(\mathbf{a}, \mathbf{a'})\rangle$ denotes the average value of $d_1(\mathbf{a}, \mathbf{a'})$ taken over all single point mutants $\mathbf{a'}$ of $\mathbf{a}$. Since $d_1(\mathbf{a}, \mathbf{a'})$ is not a true metric, we replace it by the metric $D_{\mathrm{V}}(\mathbf{a}, \mathbf{b})$ in our computation of mutational robustness. Clearly both notions are closely related.

## Implementation

In [15], McCaskill described a cubic time algorithm to compute the *partition function*

$$Z = Z(\mathbf{a}) = \sum_{s\in\mathbb{SS}(\mathbf{a})} \exp(-E(\mathbf{a},s)/RT) \tag{6}$$

for an RNA sequence $\mathbf{a} = a_1, \ldots, a_n$, where the sum is taken over all secondary structures $\mathbb{SS}(\mathbf{a})$ of $\mathbf{a}$, $E(\mathbf{a}, s)$ denotes the free energy for the structure $s$ of $\mathbf{a}$ with respect to the Turner energy parameters [13], $R$ denotes the universal gas constant and $T$ is absolute temperature. Subsequently Ding and Lawrence [16] described how to use the partition function together with a simple backtracking strategy to *sample* secondary structures of $\mathbf{a}$ from the Boltzmann ensemble of low energy structures.

If $s_0$ is a given secondary structure of length $n$, we define the *dual partition function*

$$Z^* = Z^*(s_0) = \sum_{\mathbf{a}\in\mathbb{AA}(s_0)} \exp(-E(\mathbf{a},s_0)/RT) \tag{7}$$

where the sum is taken over all RNA sequences $\mathbf{a} = a_1, \ldots, a_n$ of length $n$ that are compatible with structure $s_0$, i.e. $a_i, a_j$ constitute a Watson-Crick or wobble pair for each base pair $(i,j) \in s_0$. The set of all RNA sequences that are compatible with $s_0$ is denoted by $\mathbb{AA}(s_0)$. Note that if a sequence $\mathbf{a}$ is not compatible with the target structure $s_0$, then the energy $E(\mathbf{a}, s_0)$ is infinite, so the corresponding Boltzmann factor $\exp(-E(\mathbf{a},s_0)/RT)$ is zero and the sum in Eq. (7) could have been written over all sequences of the same length as $s_0$. Here we describe the efficient software `RNAdualPF` to compute the *dual partition function* $Z^*$ and to sample from the low energy ensemble of *sequences* that are compatible with a given secondary structure $s_0$.

### Dual partition function

If $s$ is a secondary structure on sequence $\mathbf{a} = a_1, \ldots, a_n$, then the *length* of $s$, denoted by $\ell(s)$, is equal to $n$, while the *size* of $s$, denoted by $|s|$, is the number of base pairs belonging to $s$. Similarly, if secondary structure $s$ is restricted to the interval $[i,j]$, where $1 \leq i \leq j \leq n$, then the length of the restriction of $s$ to $[i,j]$, denoted by $\ell(s[i,j])$, is equal to $j - i + 1$, while the size of the restriction of $s$ to $[i,j]$, denoted by $|s[i,j]|$, is the number of base pairs $(x,y)$ of $s$ that satisfy $i \leq x < y \leq j$.

Given an RNA sequence $\mathbf{a} = a_1, \ldots, a_n$, the McCaskill algorithm [15] computes the partition function $Z(\mathbf{a})$ defined in Eq. (6). When $\mathbf{a}$ is clear from context, $Z(\mathbf{a})$ is usually denoted by $Z$.

Given a target secondary structure $s_0$, we describe below an algorithm to compute the *dual partition function* $Z^*(s_0)$, defined as the sum of all Boltzmann factors $\exp(-E(\mathbf{a}, s_0))$, where the sum is taken over all RNA sequences $\mathbf{a} \in \mathbb{AA}(s_0)$. Unlike the McCaskill algorithm, which requires time that is cubic in the length of $\mathbf{a}$, the algorithm presented below requires time that is (essentially) linear[1] in the length of $s_0$. Our algorithm is motivated by the initialization step of the algorithm INFO-RNA [17], in which a sequence is determined, for which the free energy with respect to target structure $s_0$ is a minimum – i.e. `INFO-RNA` determines $\text{argmin}_{\mathbf{a}} E(\mathbf{a}, s_0)$.

The algorithm specification requires the notation $Z^*(i,j;x,y)$, which denotes the sum

$$Z^*(i,j;x,y) = \sum_{\mathbf{a}[i,j], a_i = x, a_j = y} \exp\left(-E\left(\mathbf{a}[i,j], s_0[i,j]\right)/RT\right) \tag{8}$$

of Boltzmann factors for sequences $\mathbf{a}[i,j] = a_i, \ldots, a_j$ for which $a_i = x, a_j = y$, and for the restriction $s_0[i,j]$, defined by

$$s_0[i,j] = \left\{(x,y) \in s_0 : i \leq x < y \leq j\right\}. \tag{9}$$

The function $Z^*(i,j;x,y)$ is defined for all base pairs $(i,j) \in s_0$; these values will be stored in an array, whose rows index base pairs of $s_0$, and whose columns are indexed by the six canonical base pairs GC, CG, AU, UA, GU, UG (see example in Table 1). Once $Z^*(i,j;x,y)$ has been computed for all base pairs that are *visible*, i.e. for which there is no base pair $(x,y)$ for which $x < i < j < y$, we can compute the full partition function $Z^*(s_0)$.

Following [17], we define a total ordering on base pairs $(i,j)$ belonging to the target structure $s_0$ that satisfy the following precedence rule for any two base pairs $(i,j)$, $(x,y)$.

$$(i,j) \prec (x,y) \Leftrightarrow x < i < j < y \text{ or } i < j < x < y \tag{10}$$

From this ordering, we assign a *base pair index* to each base pair $(i,j)$, which is defined to be the rank of $(i,j)$ in the total ordering.

The following definitions correspond to the Turner nearest neighbor energy model [13], which is an additive loop model where a loop closed by external base pair $(i,j)$ is designated as a $k$-loop, if the loop contains $k$ base pairs interior to $(i,j)$. Therefore, hairpin loops are 0-loops; base pair stacks, bulge loops and internal loops are 1-loops; and multiloops are $k$-loops for $k \geq 2$ (also called $(k+1)$-way junctions), where the additional count is due to the outer component adjacent to $(i,j)$ [18].

Since AU-base pairs that close a loop are energetically unfavorable, in the Turner energy model, there is an AU-penalty we now define:

**Table 1** Base pair dual partition function table. Given the target structure with sequence constraints depicted in Fig. 2, `RNAdualPF` computes and stores all the partial *dual partition function* values for the substructures enclosed by each base pair

| Index | i | j | Type | *AU* | *CG* | *GC* | *UA* | *GU* | *UG* | $Z^*(i,j)$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 18 | 23 | Tetraloop | 0.000 | 0.000 | 0.364 | 0.000 | 0.000 | 0.000 | 0.364 |
| 2 | 17 | 24 | Stack | 10.977 | 17.859 | 76.923 | 10.977 | 10.977 | 3.525 | 131.238 |
| 3 | 16 | 26 | R. bulge | 11.690 | 70.834 | 184.603 | 12.771 | 13.347 | 3.915 | 297.160 |
| 4 | 6 | 10 | Triloop | 0.004 | 0.010 | 0.010 | 0.004 | 0.004 | 0.004 | 0.038 |
| 5 | 5 | 11 | Stack | 0.750 | 3.022 | 5.234 | 0.899 | 0.960 | 0.256 | 11.120 |
| 6 | 3 | 13 | Int. loop | 109.842 | 256.875 | 424.976 | 108.653 | 117.851 | 108.132 | 1126.330 |
| 7 | 2 | 14 | Stack | 10853.104 | 86208.448 | 170643.321 | 12575.544 | 13285.398 | 3647.077 | 297212.891 |
| 8 | 1 | 27 | Multiloop | 1558.575 | 7895.583 | 7895.583 | 1558.575 | 1558.575 | 1558.575 | 22025.464 |
| 9 | 1 | 28 | $S_0$ | – | – | – | – | – | – | 88101.856 |

The first column indicates the *base pair index* which dictates the order in which the dual partition function is computed for different loops closed by the base pair (i,j), where we the *index* of base pair $(i,j)$ is defined to be the rank of $(i,j)$ in the total ordering defined in Eq. (10). Columns $i$ and $j$ indicate the opening and closing positions of each base pair. Type indicates the type of element in the secondary structure closed by each base pair, where R. bulge stands for right bulge, Stack for stacking base pair, and Int. loop for interior loop. The *dual partition function* $Z^*(i,j)$ of the substructure closed by base pair $(i,j)$ appears in the rightmost column, while the partition function $Z^*(i,j,X,Y)$ for each of the six canonical base pairs is given in columns 5-10. Note that for base pair 1, sequence constraints depicted in Fig. 2 force $i$ and $j$ to be instantiated respectively to G and C, hence the dual partition function $Z^*(i,j;X,Y)$ is zero for any base pair different than GC. The last column of the last row of the table shows the total dual partition function $Z^*(s_0)$ for the target structure $s_0$

$$e_{AU}(i,j,X,Y) = \begin{cases} 0.5 \text{ if } (i,j) \text{ is the outermost pair in a stem of } s_0, \text{ having AU,UA,GU,UG} \\ 0 \text{ otherwise.} \end{cases} \tag{11}$$

This AU-penalty is applied only if $(i,j)$ is a base pair adjacent to a triloop, a bulge, an internal loop or a multiloop, or if it is the outermost base pair of an external loop in target structure $s_0$, and $(i,j)$ is instantiated by one of the pairs AU, UA, GU, UG. When base-paired positions $i,j$ are clear from the context, we write $e_{AU}(X,Y)$.

Here, we assume that in parsing the input target structure, a list *BPcloseELorML* has been created of those base pairs $(i,j)$, which close either an external loop or a multiloop. Let $I$ be the indicator function, it follows that if $(i,j)$ closes an external loop or multiloop, then $\exp\left(-\frac{I[(i,j)\in BPcloseELorML]\cdot e_{AU}(X,Y)}{RT}\right)$ is the Boltzmann factor for a special AU-penalty, otherwise this factor equals 1. For clarity in the notation, this factor is denoted by $e^{(-\frac{e^{I}_{AU}(X,Y)}{RT})}$. Note that this term is distinct from the factor $\exp(-\frac{e_{AU}(X,Y)}{RT})$ applied to base pairs adjacent to a triloop, a bulge or an internal loop, which does not depend on the indicator function.

### Hairpins

Let $(i,j)$ close a hairpin in $s_0$. The hairpin free energy term $H(j-i-1)$, arising solely from entropic considerations, is defined by

$$H(j-i-1) = \begin{cases} hairpinE(j-i-1) & \text{if } j-i-1 \leq 30 \\ hairpinE(30) + 1.75RT\ln\left(\frac{j-i-1}{30}\right) & \text{otherwise} \end{cases} \tag{12}$$

where $hairpinE(j-i-1)$ designates the hairpin free energy obtained from table look-up, when $j-i-1 \leq 30$.

**Triloop** Let $TriLoop_{x,y}$ denote the collection of special triloops, *xabcy*, having an energy bonus $triloopE(xabcy)$.

$$Z^{*}(i,j;x,y) = e^{\left(-\frac{e^{I}_{AU}(x,y)}{RT}\right)} \cdot \exp\left(-\frac{H(j-i-1)+e_{AU}(xy)}{RT}\right) \times \left( \left(4^{3} - |TriLoop_{x,y}|\right) + \sum_{abc\in TriLoop_{x,y}} \exp\left(-\frac{triloopE(xabcy)}{RT}\right)\right) \tag{13}$$

**Tetraloop** Let $TetraLoop_{x,y}$ denote the collection of special tetraloops, *xabcdy*, having an energy bonus $tetraloopE(xabcdy)$. Similarly, given nucleotides $n_1, n_2 \in \mathcal{N}$, $TetraLoop_{x,y}(n_1,n_2)$ denotes the collection of special tetraloops of the form $xn_1abn_2y$. Define $Z^{*}(i,j;x,y)$ by

$$Z^{*}(i,j;x,y) = e^{\left(-\frac{e^{I}_{AU}(x,y)}{RT}\right)} \cdot \exp\left(-\frac{H(j-i-1)}{RT}\right) \times \sum_{n_1,n_2\in\mathcal{N}} \left( \exp\left(-\frac{mismatch(x,y,n_1,n_2)}{RT}\right) \times \left\{ \left(4^{2} - |TetraLoop_{x,y}(n_1,n_2)|\right) + \sum_{ab\in TetraLoop_{x,y}(n_1,n_2)} \exp\left(-\frac{tetraloopE(xn_1abn_2y)}{RT}\right)\right\}\right) \tag{14}$$

**Hexaloop** Let $HexaLoop_{x,y}$ denote the collection of special hexaloops, *xabcdefy*, having an energy bonus $hexaloopE(xabcdefy)$. Similarly, given nucleotides $n_1, n_2$, $HexaLoop_{x,y}(n_1,n_2)$ denotes the collection of special hexaloops of the form $xn_1abcdn_2y$. Define $Z^{*}(i,j;x,y)$ by

$$Z^{*}(i,j;x,y) = e^{\left(-\frac{e^{I}_{AU}(x,y)}{RT}\right)} \cdot \exp\left(-\frac{H(j-i-1)}{RT}\right) \times \sum_{n_1,n_2\in\mathcal{N}} \left( \exp\left(-\frac{mismatch\,(x,y,n_1,n_2)}{RT}\right) \times \left\{ \left(4^{4} - |HexaLoop_{x,y}\,(n_1,n_2)\,|\right) + \sum_{ab\in HexaLoop_{x,y}(n_1,n_2)} \exp\left(-\frac{HexaloopE\,(xn_1abcdn_2y)}{RT}\right)\right\}\right) \tag{15}$$

**Hairpin size exceeds four and is different than six** Define $Z^{*}(i,j;x,y)$ by

$$Z^{*}(i,j;x,y) = e^{\left(-\frac{e^{I}_{AU}(x,y)}{RT}\right)} \cdot \exp\left(-\frac{H(j-i-1)}{RT}\right) \times \left(\sum_{n_1,n_2\in\mathcal{N}} \exp\left(-\frac{mismatch\,(x,y,n_1,n_2)}{RT}\right) \cdot 4^{j-i-3}\right) \tag{16}$$

### Stacked base pairs, bulges and internal loops

Here, we consider the case of a 1-loop, which comprises the case of stacked base pairs, bulges and internal loops. The following cases correspond to each possibility.

**Stacked base pair** In this case, $(i,j)$ stacks on the base pair $(i+1,j-1)$, and the partition function $Z^{*}(i+1,j-$

$1; U, V)$ has been computed. Let $stack(X, Y, U, V)$ denote the free energy of base stack $\begin{matrix} 5' - \mathrm{XU} - 3' \\ 3' - \mathrm{YV} - 5' \end{matrix}$ obtained by table look-up.

$$Z^*(i,j;X,Y) = e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)} \cdot \sum_{UV \in \mathcal{B}} \exp\left(-\frac{stack(X,Y,U,V)}{RT}\right) \times Z^*\left(i+1, j-1, U, V\right) \tag{17}$$

**Bulge loop** In this case, $(i,j)$ closes a bulge in $s_0$. Since bulge size may exceed the values in table look-up, we define the free energy for a bulge of size $r$ by

$$bulge(r) = \begin{cases} bulgeE(r) & \text{if } r \leq 30 \\ bulgeE(30) + 1.75RT \ln\left(\frac{r}{30}\right) & \text{otherwise.} \end{cases} \tag{18}$$

If $(i,j)$ closes a left bulge of size $r$ in $s_0$, then the bulge is closed by base pair $(i+r+1, j-1)$ involving nucleotide pair $U, V$, and

$$Z^*(i,j;X,Y) = e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)} \cdot \sum_{UV \in \mathcal{B}} \exp\left(-\frac{e_{AU}\left(i,j,X,Y\right)}{RT}\right) \times \exp\left(-\frac{bulge(r)}{RT}\right) \cdot 4^r \cdot Z^*\left(i+r+1, j-1, U, V\right) \tag{19}$$

while if $(i,j)$ closes a right bulge in $s_0$, then the bulge is closed by base pair $(i+1, j-r-1)$ involving nucleotide pair $U, V$, and

$$Z^*(i,j;X,Y) = e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)} \cdot \sum_{UV \in \mathcal{B}} \exp\left(-\frac{e_{AU}(i,j,X,Y)}{RT}\right) \times \exp\left(-\frac{bulge(r)}{RT}\right) \cdot 4^r \cdot Z^*\left(i+1, j-r-1, U, V\right) \tag{20}$$

**Internal loop** In this case, $(i,j)$ closes an internal loop in $s_0$, whose left [resp. right] portion is of size $r_1$ [resp. $r_2$]. Since internal loop size $r = r_1 + r_2$ may exceed the values in table look-up, we define the free energy for an internal loop of size $r$ by

$$internal(r) = \begin{cases} internalE(r) & \text{if } r \leq 30 \\ internalE(30) + 1.75RT \ln\left(\frac{r}{30}\right) & \text{otherwise.} \end{cases} \tag{21}$$

The closing base pair $(i+r_1+1, j-r_2-1)$ of the internal loop of size $r = r_1 + r_2$ may involve the nucleotides $UV \in \mathcal{B}$, while the unpaired (mismatch) nucleotides in positions $i+1, j-1, i+r_1, j-r_2$ may involve $A, B, C, D \in \mathcal{N}$. In addition, there is an energy penalty for non symmetric internal loops, $min(asym \cdot |r_1 - r_2|, maxAsym)$, where the value of the constants *asym* and *maxAsym* are given in the Turner energy model. Thus

$$Z^*(i,j;X,Y) = e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)} \cdot \exp\left(-\frac{min\left(asym \cdot |r_1 - r_2|, maxAsym\right)}{RT}\right) \times \sum_{UV \in \mathcal{B}} \sum_{A,B,C,D \in \mathcal{N}} \exp\left(-\frac{e_{AU}\left(i,j,X,Y\right)}{RT}\right) \times \exp\left(-\frac{internal\left(r_1 + r_2\right)}{RT}\right) \cdot 4^{r_1+r_2-4} \times \exp\left(-\frac{mismatch(X,Y,A,B) + mismatch(V,U,D,C)}{RT}\right) \times Z^*\left(i+r_1+1, j-r_2-1, U, V\right) \tag{22}$$

### *External loop*

Despite the fact that, by following the total order on base pairs defined in Eq. (10), the *dual partition function* of multiloops is always computed before the *dual partition function* of the external loop, the computation of the *dual partition function* of multiloops will be easier to understand if the *dual partition function* of the external loop is defined in advance.

In order to improve speed, some implementations of RNA thermodynamics-based algorithms ignore the contribution of dangling positions, which corresponds to `Vienna RNA Package -d0` flag. `RNAdualPF` also includes this option, which dramatically increases the speed of the algorithm. The reason behind this difference of performance is clear from the following definitions.

Suppose that $H = [(i_1, j_1), \ldots, (i_k, j_k)]$ constitutes the list of $k$ external base pairs of $s_0$, where $i_1 < j_1 < i_2 < j_2 < \cdots < i_k < j_k$. For each $(i_r, j_r)$, with $1 \leq r \leq k$, and for each choice of base pair GC, CG, AU, UA, GU, UG, the value $Z^*(i_r, j_r; X_r, Y_r)$ has been previously computed and stored by dynamic programming, as well as the sum $Z^*(i_r, j_r)$. When the contribution of dangles is ignored, the *dual partition function* of an external loop with $\ell$ nucleotide positions external to every base pair is defined by

$$Z^*(s_0) = 4^{\ell} \cdot \prod_{r=1}^{k} Z^*(i_r, j_r) \tag{23}$$

where $\ell = n - \sum_{r=1,\ldots,k} (j_r - i_r + 1)$ and $n$ is the length of the target structure $s_0$.

The default treatment of dangles in `RNAdualPF` described below corresponds to `Vienna RNA Package -d2` flag, where both flanking positions of each external base pair contribute to the free energy. Let $D = [a_1, b_1, \ldots, a_k, b_k] \subseteq [i_1 - 1, j_1 + 1, \cdots, i_k, j_k]$ be a list of those nucleotide positions that are adjacent to the $k$ external base pairs $(i_1, j_1), \ldots, (i_k, j_k)$. The ordered multiset $[a_1, b_1, \ldots, a_k, b_k]$ can be considered as a collection of constraints, so that (for instance) if $a_2 = i_2 - 1$, and $a_2 = j_1 + 1$, then $a_2 = b_1$ and any nucleotide value that is assigned to $b_1$ must simultaneously be assigned to $a_2$. Moreover, there can also be an overlap between the list of base paired positions in $H\ [i_1, j_1, \ldots, i_k, j_k]$ and the multiset $D = [a_1, b_1, \ldots, a_k, b_k]$. If (for instance) $j_1 = i_2 - 1$, then $b_1 = i_2$ and $a_2 = j_1$. Therefore, in the computation we have to account for these constraints. Let $m$ denote the number of unpaired positions in $D$, without repetitions, and define $A_r, B_r$ as the nucleotides instantiated respectively at $a_r, b_r$. The energy term for a $5'$-dangle [resp. $3'$-dangle] on base pair $(x, y)$ with nucleotides $U, V$ is denoted by $E_{d5}(x, y, x-1; U, V, W)$ [resp $E_{d3}(x, y, y+1; U, V, W)$] where the dangle position $x - 1$ [resp. $y + 1$] is assigned nucleotide $W$. With the notation just described, we have

$$Z^*(s_0) = \sum_{\langle (U_1,V_1),\ldots,(U_k,V_k)\rangle \in \mathcal{B}^k} \ \sum_{\{A_1,B_1,\ldots,A_k,B_k \in \mathcal{N}^{2k}\}} 4^{\ell - m} \times \prod_{r=1}^{k} \left(Z^*\left(i_r, j_r; U_r, V_r\right)\right. \times \exp\left(-\frac{E_{d5}\left(i_r, j_r, a_r; U_r, V_r, A_r\right) + E_{d3}\left(i_r, j_r, b_r; U_r, V_r, B_r\right)}{RT}\right) \tag{24}$$

Depending on the target structure $s_0$, it can happen that the second sum of Eq. (24) must be restricted to range over strictly less than $4^{2k}$ many RNA sequences. This is explained as follows. If $i_1 = 1$ [resp. $j_r = n$] then there is no position for a $5'$ [resp. $3'$] dangle, and hence the nucleotide sequences considered in the second summation would have length strictly less than $2k$. Moreover, certain $5'$ dangled positions could be identical to $3'$ dangle positions, which arises for instance when $j_k + 2 = i_{k+1}$; alternatively, certain dangled positions could be identical with base-paired positions, which arises for instance when $j_k + 1 = i_{k+1}$. In such situations, instantiations of the $3'$-dangle on $(i_k, j_k)$ and the $5'$-dangle on $(i_{k+1}, j_{k+1})$ are not independent, thus leading to a restriction of the range of the second summation in Eq. (24). A similar restriction is implicitly assumed in the treatment of external loops in this section and of multiloops in the next section.

The algorithm performance can be improved by dividing the external loop into groups of components having interdependently constrained dangling positions, as just explained. Define two base pairs $(x, y), (x', y')$ as adjacent if $x < y < x' < y'$ and $x' - y \leq 2$ – i.e. dangling positions of the base pairs $(x, y), (x', y')$ are constrained. Let $G$ denote a *maximal* collection of *adjacent* base pairs belonging to $H = [(i_1, j_1), \ldots, (i_k, j_k)]$, together with their associated dangle positions in $D = [i_1 - 1, j_1 + 1, \ldots, i_k - 1, j_k + 1]$. It is important to note that $H \cup D$ is thus partitioned into a collection of $g$ disjoint groups $\mathcal{G} = [G_1, \ldots, G_g]$. Therefore, we can divide an external loop of $k$ helices into a collection groups $\mathcal{G}$ of size $g \leq k$, and $p$ unpaired positions that are external to every base pair of $s_0$ and not adjacent to any base pair.

For a group $G$ with $h$ base pairs, let $H(G) = [(\kappa_1, \lambda_1), \ldots, (\kappa_k, \lambda_k)]$ denote the list of base pairs in $G$, and let $D(G) = [\alpha_1, \beta_1, \ldots, \alpha_h, \beta_h] \subseteq [\kappa_1 - 1, \lambda_1 + 1, \cdots, \kappa_h - 1, \lambda_h + 1]$ denote their associated dangle positions. If $U_r, V_r, A_r, B_r$ denote the nucleotides instantiated at the base pair $r = (\kappa_r, \lambda_r)$ and its respective dangling positions $\alpha_r, \beta_r$ respectively, then the *dual partition function* of $G$ is the following.

$$Z^*(G) = \sum_{\langle (U_1,V_1),\ldots,(U_h,V_h)\rangle \in \mathcal{B}^h} \ \sum_{\{A_1,B_1,\ldots,A_h,B_h \in \mathcal{N}^{2h}\}} \times \prod_{r=1}^{h} \left(Z^*\left(\kappa_r, \lambda_r; U_r, V_r\right)\right. \times \exp\left(-\frac{E_{d5}\left(\kappa_r, \lambda_r, \alpha_r; U_r, V_r, A_r\right) + E_{d3}\left(\kappa_r, \lambda_r, \beta_r; U_r, V_r B_r\right)}{RT}\right) \tag{25}$$

where the range of the second summation can be constrained by the overlap among positions in $D(G)$ and between positions in $D(G)$ and $H(G)$, as explained for Eq. (24).

Finally, since there are no shared dangling positions between groups, the *dual partition function* of an external loop is defined by

$$Z^*(s_0) = 4^p \cdot \prod_{r=1}^{g} Z^*(G_r). \tag{26}$$

### *Multiloop*

Suppose that $(i, j)$ closes a multiloop in $s_0$, which is a $k$-loop, or $(k+1)$-way junction, for $k > 1$, where there are $\ell$ unpaired bases in the multiloop. Suppose that the $k$ components of the multiloop are closed by the base pairs $(i_1, j_1), \ldots, (i_k, j_k)$ with the property that $i < i_1 < j_1 < i_2 < j_2 < \cdots < i_k < j_k < j$. Assume that for all nucleotide choices in $\mathcal{B}$ for each of the $k$ base pairs of the multiloop $(i_r, j_r)$, for $1 \leq r \leq k$, the value $Z^*(i_r, j_r; X_r, Y_r)$ has previously been computed and stored by dynamic programming, as well as the sum $Z^*(i_r, j_r)$. The computation of the *dual partition function* is similar to that of the external loop. However, in this case we have to add the contribution of the base pair closing the multiloop $(i, j)$, the AU-penalties applied to this base pair, and the energetic penalty of a multiloop $a + b \cdot (k+1) + c \cdot \ell$, where the values of the constants $a$, $b$ and $c$ are given in the Turner energy model. Then, the *dual partition function* of a multiloop without accounting for dangling positions is

$$Z^*(i, j; X, Y) = e^{\left(-\frac{e^{I}_{AU}(X,Y)}{RT}\right)} \cdot \exp\left(-\frac{a + b \cdot (k+1) + c\ell}{RT}\right) \cdot 4^{\ell} \times \exp\left(-\frac{e_{AU}(i, j, X, Y)}{RT}\right) \cdot \sum_{\langle (U_1,V_1),\ldots,(U_k,V_k)\rangle \in \mathcal{B}^k} \times \prod_{r=1}^{k} Z^*\left(i_r, j_r; U_r, V_r\right) \tag{27}$$

The notation we use to define the *dual partition function* of multiloops with dangling positions is similar to that described for external loops. However, some modifications are required in the previously given definitions, since we have to take into account the flanking positions of the base pair $(i,j)$ closing the multiloop. Let $H = [(i_1,j_1),\ldots,(i_k,j_k),(i,j)]$ be the collection of $k$ base pairs closing one of the $k$ components of the multiloop, and the base pair $(i,j)$ closing the multiloop, and define the multiset $D = [a_1,b_1,\ldots,a_{k+1},b_{k+1}] \subseteq [i_1-1,j_1+1,\cdots,i_k-1,j_k+1,i+1,j-1]$ of nucleotide positions adjacent to the base pairs in $H$. Due to the possible overlap with the base pair closing the multiloop and its flanking positions, there are additional constraints in the ordered multiset $[a_1,b_1,\ldots,a_{k+1},b_{k+1}]$, so that (for instance) if $a_1 = i_1-1$, and $i_1 = i+1$, then $a_1 = a_{k+1}$ and any nucleotide value that is assigned to $a_1$ must simultaneously be assigned to $a_{k+1}$. Moreover, there can also be an overlap between the list of base paired positions $[i_1,j_1,\ldots,i_k,j_k,i,j]$ and the multiset $[a_1,b_1,\ldots,a_{k+1},b_{k+1}]$. If (for instance) $i = i_1-1$, then $a_{k+1} = i_1$ and $a_1 = i$.

Let $m$ denote the number of unpaired positions in $D$, without repetitions. Then, the *dual partition function* of a multiloop with dangling positions is defined as follows.

$$\begin{aligned} Z^*(i,j;X,Y) = e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)} \cdot & \sum_{\langle (U_1,V_1),\ldots,(U_k,V_k)\rangle \in \mathcal{B}^k} \; \sum_{\{A_1,B_1,\ldots,A_{k+1},B_{k+1} \in \mathcal{N}^{2(k+1)}\}} \\ & \exp\left(-\frac{a+b\cdot(k+1)+c\ell}{RT}\right) \cdot 4^{\ell-m} \cdot \exp\left(-\frac{e_{AU}\left(i,j,X,Y\right)}{RT}\right) \\ & \times \prod_{r=1}^{k} \left( Z^*\left(i_r,j_r;U_r,V_r\right) \right. \\ & \times \exp(-\frac{E_{d5}\left(i_r,j_r,a_r;U_r,V_r,A_r\right)+E_{d3}(i_r,j_r,b_r;U_r,V_r,B_r)}{RT}\Bigg) \\ & \times \exp\left(-\frac{E_{d3}\left(j,i,a_{k+1};Y,X,A_{k+1}\right)+E_{d5}\left(j,i,b_{k+1};Y,X,B_{k+1}\right)}{RT}\right) \end{aligned} \tag{28}$$

As explained for Eq. (24), it can happen that the second summation must be restricted to range over strictly less than $4^{2k}$ many RNA sequences.

A decomposition similar that for external loops can be performed to improve the performance in the computation of the *dual partition function* of a multiloop. In a multiloop, in addition to the adjacency definition given for external loops, we consider the base pair $(i,j)$ that closes the multiloop as adjacent to a base pair $(x,y)$ that closes a component of the multiloop, where $i < x < y < j$, if either $x \leq i+2$ or $y \geq j-2$. Then, let G denote a *maximal* collection of *adjacent* base pairs belonging to $H = [(i_1,j_1),\ldots,(i_k,j_k),(i,j)]$, together with their associated dangle positions in $D = [i_1-1,j_1+1,\ldots,i_k-1,j_k+1,i+1,j-1]$. This decomposition produces a collection $\mathcal{G}$ of $g$ disjoint groups $G_1,\ldots,G_g$, one of which, designated the *closing group* $G_c$ contains the closing base pair $(i,j)$ of the multiloop, and $g-1$ of which, designated as *non-closing groups* $G_{nc}$, do not contain the base pair $(i,j)$.

*Non-closing groups* have the same composition as those defined for external loops – i.e. a collection of $h$ base pairs $H(G_{nc}) = [(\kappa_1,\lambda_1),\ldots,(\kappa_h,\lambda_h)]$ and a set of dangling positions $D(G_{nc}) = [\alpha_1,\beta_1,\ldots,\alpha_h,\beta_h] \subseteq [\kappa_1-1,\lambda_1+1,\cdots,\kappa_h-1,\lambda_h+1]$. Therefore, we can compute the *dual partition function* $Z(G_{gc})$ of a *non-closing group* as described in Eq. (25). In addition, the collection of *non-closing groups* of size $g-1$ of a multiloop of $k$ components is denoted by $\mathcal{G}_{nc}$, where $0 \leq (g-1) \leq k$.

Therefore, a multiloop of $k$ components and $\ell$ unpaired positions can be decomposed into one closing group $G_c$, a collection of non-closing groups $\mathcal{G}_{nc}$, and $p$ unpaired positions that are not adjacent to any base pair, with $0 \leq p \leq \ell$.

In a *non-closing group*, the collection of base pairs of size $h+1$ is denoted by $H(G_c) = [(\kappa_1,\lambda_1),\ldots,(\kappa_h,\lambda_h),(i,j)]$, where the base pair $(i,j)$ closing the multiloop is at the last position. The ordered multiset of adjacent positions is denoted by $D(G_c) = [\alpha_1,\beta_1,\ldots,\alpha_{h+1},\beta_{h+1}] \subseteq [\kappa_1-1,\lambda_1+1,\cdots,\kappa_h-1,\lambda_h+1,i+1,j-1]$, where the positions adjacent to $i$ and $j$ are at the last positions are respectively denoted by $\alpha_{h+1},\beta_{h+1}$. A graphical example of a *closing group* and a *non-closing group* is shown in Fig. 1e, where the positions of a *non-closing group* with 1 base pair are highlighted in green and the positions of the *closing group* are highlighted in red and blue, and where the base pair $(i,j)$ that closes the multiloop is depicted in red.

For a *closing group* $G_c$ with $h+1$ base pairs in $H(G_c) = [(\kappa_1,\lambda_1),\ldots,(\kappa_h,\lambda_h),(i,j)]$ and their flanking positions $D(G_c) = [\alpha_1,\beta_1,\ldots,\alpha_{h+1},\beta_{h+1}] \subseteq [\kappa_1-1,\lambda_1+1,\cdots,\kappa_h-1,\lambda_h+1,i+1,j-1]$, let $X,Y$ denote the nucleotides assigned to the closing base pair of the multiloop $(i,j)$, and let $U_r,V_r,A_r,B_r$ denote the nucleotides assigned respectively to the base pair $r = (\kappa_r,\lambda_r,)$ and its flanking positions $\alpha_r,\beta_r$. Then, the the *dual partition function* $Z^*(G_c;X,Y)$ of the *closing group* is defined by

$$\begin{aligned} e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)} \cdot & \sum_{\langle (U_1,V_1),\ldots,(U_k,V_h)\rangle \in \mathcal{B}^h} \; \sum_{\{A_1,B_1,\ldots,A_{h+1},B_{h+1} \in \mathcal{N}^{2(h+1)}\}} \\ & \times \exp\left(-\frac{e_{AU}\left(i,j,X,Y\right)}{RT}\right) \cdot \prod_{r=1}^{h} \left(Z^*\left(\kappa_r,\lambda_r;U_r,V_r\right)\right. \\ & \times \exp\left(-\frac{E_{d5}\left(\kappa_r,\lambda_r,\alpha_r;U_r,V_r,A_r\right)+E_{d3}\left(\kappa_r,\lambda_r,\beta_r;U_r,V_r,B_r\right)}{RT}\right) \\ & \times \exp\left(-\frac{E_{d3}\left(j,i,\alpha_{h+1};Y,X,A_{h+1}\right)+E_{d5}\left(j,i,\beta_{h+1};Y,X,B_{h+1}\right)}{RT}\right) \end{aligned} \tag{29}$$

In the same way as in Eq. (24), the values of the second summation are constrained to the possible choices among overlapping positions.

Then, the *dual partition function* $Z^*(i,j;X,Y)$ of the multiloop with $k$ components and $\ell$ unpaired positions,

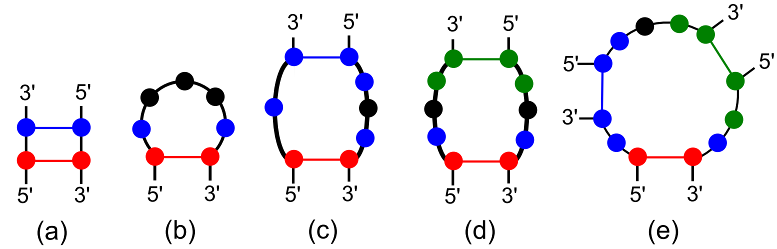

**Fig. 1** Sampling dependency examples in RNAdualPF for different structural elements: (**a**) stacked base pair, (**b**) hairpin, (**c**) 1 × 3 internal loop, (**d**) 3 × 3 internal loop and (**e**) multiloop. Base pair $(i,j)$ to be sampled is highlighted in *red*, positions whose energy contribution is dependent on the instantiation of $(i,j)$ are highlighted in *blue*, and positions that are mutually dependent, but independent of the instantiation of $(i,j)$, are highlighted in *green*. Unpaired positions where the nucleotide choice has no effect in the free energy of the structure are indicated in *black*

where $p$ of which are not adjacent to any base pair, is defined by

$$Z^*(i,j;X,Y) = \exp\left(-\frac{a+b\cdot(k+1)+c\ell}{RT}\right)\cdot 4^p \quad (30)$$
$$\times Z^*(G_c;X,Y)\cdot \prod_{G_{nc}\in\mathcal{G}_{nc}} Z^*(G_{nc})$$

### Sampling

Once the *dual partition function* $Z^*(i,j)$ and its subcases $Z^*(i,j;X,Y)$ for each base pair $(i,j)$ have been computed, it is possible to perform a Boltzmann weighted sampling of positions $i$ and $j$. For example, given the target structure with sequence constraints depicted in Fig. 2, RNAdualPF computes the *dual partition function* table shown in Table 1. The *dual partition function* of the substructure enclosed by the base pair $(i,j)$ is $Z^*(i,j)$, and the *dual partition function* of the substructure enclosed by the base pair $(i,j)$ where $i,j$ are currently instantiated by the nucleotides $X,Y$ is denoted by is $Z^*(i,j;X,Y)$. Therefore, the Boltzmann probability of $X,Y$ at positions $i,j$ in the substructure enclosed by the base pair $(i,j)$ is $Z^*(i,j;X,Y)/Z^*(i,j)$ and can be sampled using the roulette wheel method.

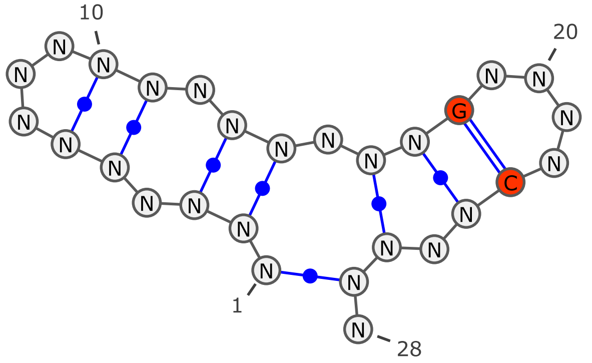

**Fig. 2** Target structure with sequence constraints used as input of RNAdualPF to compute the *dual partition function* values shown in Table 1. Sequence constraints are highlighted in *red*

Due to the Turner energy model, it is necessary to determine nucleotide positions whose instantiation influences the energy (hence Boltzmann probability) of other positions, and subsequently all mutually dependent positions must be instantiated simultaneously. Figure 1 illustrates the mutual dependencies that must be considered when sampling different types of elements, where the base pair $(i,j)$ to be sampled is highlighted in red, positions whose sampling probability is dependent on the instantiation of $(i,j)$ are highlighted in blue, and positions that are mutually dependent, but independent of the instantiation of $(i,j)$, are highlighted in green.

Since the dynamic programming algorithm for the *dual partition function* proceeds from inner to outer base pairs, using the total ordering $\prec$ in Eq. (10), the sampling order of base pairs proceeds from outer to inner positions, i.e. from largest *base pair index* to smallest. In order to account for mutual dependencies in the sampling step, we define the function *sample*$(k,T,i,j,X,Y)$ for each base pair $(i,j)$ in $S_0$, where $k$ indicates the *base pair index* defined from Eq. (10), $T$ indicates the type of structural element closed by base pair $(i,j)$ in the target RNA secondary structure, as shown in Table 1, and $X,Y$ are the instantiated nucleotides at positions $(i,j)$. Due to the mutual dependencies, sampling a base pair with *base pair index* $k$ closing an $m$-loop, for $m>0$, forces the instantiation of all inner closing base pairs of the $m$-loop, and the *base pair index* of each such inner base pair is strictly less than $k$. For this reason, except in the case of external loops, the outermost base pair $(i,j)$ has been always instantiated before *sample*$(k,T,i,j,X,Y)$ is called, and therefore the instantiation $X,Y$ is given as a parameter of the sampling function.

The Boltzmann probability of each possible instantiation of mutually dependent positions can be computed on the fly in the backward step. However, in order to improve the speed of the algorithm, in the forward step RNAdualPF stores (for each base pair) the conditional *dual partition function* values of instantiations of interdependent positions. These tables are used by the sampling function, since each value corresponds to the *dual partition function* conditional on a specific instantiation of the positions to be sampled by *sample*$(k, T, i, j, X, Y)$. Since the sampling procedure depends on the type $T$ of element, we describe the function *sample*$(k, T, i, j, X, Y)$ for each type of element – hairpin, stacked base pair, internal loop (which also comprises left and right bulge), multiloop and external loop as depicted in Fig. 1. For each of these cases, the values are stored in the conditional *dual partition function* table associated with the closing base pair $(i, j)$.

### *Hairpins*

When hairpin size exceeds three (Fig. 1a), since the base pair $(i, j)$ has been previously instantiated, flanking positions $i + 1, j - 1$ are sampled first. Given the current assignment $X, Y$, the Boltzmann probability of sampling respectively the nucleotides $U, V$ at the flanking positions $i + 1, j - 1$ is

$$\begin{aligned} &P\left(i+1=U, j-1=V | i=X, j=Y\right) \\ &\quad= \frac{Z^{*}\left(i,j,i+1,j-1; X,Y,U,V\right)}{Z^{*}\left(i,j; X,Y\right)} \end{aligned} \tag{31}$$

Therefore, in the forward step RNAdualPF stores in a table the conditional *dual partition function* of each possible instantiation $\{X, Y, U, V\}$ of the base pair $(i, j)$ and its flanking positions $i + 1, j - 1$ respectively, defined by

$$\begin{aligned} &Z^{*}\left(i,j,i+1,j-1; X,Y,U,V\right) \\ &\quad= e^{\left(-\frac{e^{I}_{AU}(X,Y)}{RT}\right)} \cdot \exp\left(-\frac{H(j-i-1)}{RT}\right) \cdot \\ &\qquad \exp\left(-\frac{mismatch(X,Y,U,V)}{RT}\right) \cdot 4^{j-i-3} \end{aligned} \tag{32}$$

Then, remaining unpaired positions are uniformly sampled, since the nucleotide choice does not change the final free energy. Triloops, tetraloops and hexaloops are exceptions to this rule, since there are special loops that contribute to or penalize the free energy. In those cases, we have to account for the special loops, as defined in "Hairpins" section.

Although it could seem to be a waste of space to store a different conditional *dual partition function* table for each base pair $(i, j)$, even for two different hairpins of the same size in the target structure, one should note that RNAdualPF allows sequence constraints, and thus $Z^{*}(i, j)$ could possibly differ from $Z^{*}(i', j')$ when $(i, j)$ and $(i', j')$ close hairpins of the same size.

### *Stacking base pairs*

As depicted in Fig. 1b, sampling probability of a base pair with *base pair index* $k - 1$ is dependent on the value sampled at the adjacent stacking base pair with *base pair index* $k$. Therefore, *sample*$(k, Stack, i, j, X, Y)$ samples the base pair $(i + 1, j - 1)$ using the conditional probability given the instantiation of base pair $(i, j)$ by $X, Y$, defined as follows:

$$\begin{aligned} &P\left(i+1=U, j-1=V | i=X, j=Y\right) \\ &\quad= \frac{Z^{*}\left(i,j,i+1,j-1; X,Y,U,V\right)}{Z^{*}\left(i,j; X,Y\right)} \end{aligned} \tag{33}$$

The conditional *dual partition function* values stored in the forward step correspond to each instantiation $\{X, Y, U, V\}$ of the base pairs $(i, j)$, $(i+1, j-1)$, denoted by

$$\begin{aligned} &Z^{*}\left(i,j,i+1,j-1; X,Y,U,V\right) \\ &\quad= e^{\left(-\frac{e^{I}_{AU}(X,Y)}{RT}\right)} \cdot \exp\left(-\frac{stack(X,Y,U,V)}{RT}\right) \\ &\qquad \times Z^{*}\left(i+1,j-1,U,V\right) \end{aligned} \tag{34}$$

### *Internal loops*

The energy contribution of internal loops in the Turner energy model depends on the flanking unpaired positions of both the inner and outer closing base pairs, hence the sampling probability of the inner base pair cannot be separated from the adjacent unpaired positions. Moreover, for specific sizes of internal loop ($1 \times 1$, $1 \times 2$, $2 \times 1$, $1 \times N$ and $N \times 1$), the inner and outer closing base pairs share flanking positions. In these cases, all the unpaired positions and the outer base pair must be sampled at the same time, since the energy contribution of each combination of base pairs and flanking positions is different. In the $1 \times 3$ internal loop depicted in Fig. 1c, if the outer base pair $(i, j)$ is instantiated by $X, Y$, then let $P(k = U, l = V, n_1 = A, n_2 = B, n_3 = C | i = X, j = Y)$ denote the probability of sampling the nucleotides $U, V, A, B, C$ respectively at positions $k, l, n_1, n_2, n_3$, where $(k, l)$ is the inner closing base pair, $n_1$ is the flanking position at $i + 1$ shared by the base paired positions $i$ and $k$, and $n_2$ and $n_3$. In the following equation, let $P(U, V, A, B, C|X, Y)$ abbreviate the conditional probability just defined. Then

$$\begin{aligned} &P\left(U,V,A,B,C|X,Y\right) \\ &\quad= \frac{Z^{*}\left(i,j,k,l,n_1,n_2,n_3; X,Y,U,V,A,B,C\right)}{Z^{*}\left(i,j; X,Y\right)} \end{aligned} \tag{35}$$

RNAdualPF computes and stores the conditional *dual partition function* of each possible instantiation $\{X, Y, U, V, A, B, C\}$ respectively at positions $i, j, k, l, n_1, n_2, n_3$, where the value $Z^*(i, j, k, l, n_1, n_2, n_3; X, Y, U, V, A, B, C)$ is defined by

$$e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)} \cdot \exp\left(-\frac{min(asym \cdot |(k-i)-(j-l)|, maxAsym)}{RT}\right)$$
$$\times 4^{j-l-3} \cdot \exp\left(-\frac{e_{AU}(i,j,X,Y)}{RT}\right)$$
$$\times \exp\left(-\frac{internal\left(k-i+j-l-2\right)}{RT}\right)$$
$$\times \exp\left(-\frac{mismatch(X,Y,A,B) + mismatch(V,U,C,A)}{RT}\right)$$
$$\times Z^*(k,l,U,V) \tag{36}$$

For internal loops of sizes ($1 \times 1$, $1 \times 2$, $2 \times 1$, $1 \times N$ and $N \times 1$) similar conditional *dual partition function* tables are computed following the definitions in "Internal loop" section.

**Other internal loops:** When there are no shared flanking positions between the two base pairs that close an internal loop, as depicted in Fig. 1d, the energy contribution of innermost base pair and its respective flanking positions is independent of those of the outermost base pair.

In this case, RNAdualPF samples first the flanking positions $i+1, j-1$ of the outermost base pair $(i, j)$, whose sampling probability is solely dependent on the instantiated nucleotides $X, Y$ at positions $i, j$. Is not necessary to store any conditional *dual partition function* for sampling these positions, since the probability of sampling the values $A, B$ at the flanking positions $i+1, j-1$, given the assignment $X, Y$ is defined by

$$P\left(i+1=A, j-1=B | i=X, j=Y\right) = \frac{\exp\left(-\frac{mismatch(X,Y,A,B)}{RT}\right)}{\sum_{C,D\in\mathcal{N}} \exp\left(-\frac{mismatch(X,Y,C,D)}{RT}\right)} \tag{37}$$

where mismatch penalties are obtained from table lookup. Finally, the innermost base pair $(k, l)$ and its flanking positions $k-1, l+1$ are sampled together. In this case, we need to store an additional value $Z^*(k-1, l+1)$, which is given by

$$Z^*\left(k-1, l+1\right) = \sum_{UV\in\mathcal{B}} \sum_{C,D\in\mathcal{N}} \exp\left(-\frac{mismatch(V,U,D,C)}{RT}\right) \cdot Z^*(k,l,U,V) \tag{38}$$

Then, following the same notation, the probability of sampling the nucleotides $V, U, D, C$ respectively at positions $k, l, k-1, l+1$ is

$$P\left(k=V, l=U, k-1=D, l+1=C\right) = \frac{Z^*\left(k,l,k-1,l+1; V,U,D,C\right)}{Z^*\left(k-1,l+1\right)} \tag{39}$$

Therefore, the conditional *dual partition function* of each possible instantiation $\{V, U, D, C\}$ stored in the corresponding table is defined as

$$Z^*\left(k,l,k-1,l+1; V,U,D,C\right) = \exp\left(-\frac{mismatch(V,U,D,C)}{RT}\right) \cdot Z^*(k,l,U,V) \tag{40}$$

Finally, since the remaining unpaired position does not contribute to the free energy, it is uniformly sampled.

#### *Multiloops and external loops*

As explained in "External loop" section, if dangling positions are not included in the computation, sampling an external base pair or the closing base pair $(i, j)$ of a multiloop from $Z^*(i, j)$ is trivial. On the other hand, by including dangling positions in the sampling, there is a dramatic increase in the space complexity of RNAdualPF, albeit the space used is only a constant factor larger. However, the decompositions into groups described in "External loop" and "External loop" sections allow to sample the positions of each group independently.

The example shown in Fig. 1e depicts a multiloop with two groups: a *non-closing group* $G_{nc}$ highlighted in green, and a *closing group* $G_c$ highlighted in red and blue, where the closing base pair of the multiloop $(i, j)$ is marked in red.

In a *non-closing group* $G_{nc}$ all base pairs in $H(G_{nc})$ and dangling positions in $D(G_{nc})$ must be sampled together. Therefore, the conditional *dual partition function* of each possible instantiation of nucleotides at the $h$ closing pairs in $H(G_{nc})$ and their adjacent positions in $D(G_{nc})$ is stored. Let $\mathcal{U} = \{U_1, V_1, \ldots, U_h, V_h\}$ denote an instantiation of the $h$ base pairs in $H(G_{nc}) = [\kappa_1, \lambda_1, \ldots, \kappa_h, \lambda_h]$, and let $\mathcal{W} = \{A_1, B_2, \ldots, A_h, B_h\}$ denote an instantiation of the $h$ flanking positions in $D(G_{nc}) = [\alpha_1, \beta_1, \ldots, \alpha_h, \beta_h]$ in the *non-closing group* $G_{nc}$. Then, the probability of sampling $\mathcal{U}, \mathcal{W}$ is

$$P(H(G_{nc}) = \mathcal{U}, D(G_{nc}) = \mathcal{W}) = \frac{Z^*\left(G, H\left(G_{nc}\right), D\left(G_{nc}\right); \mathcal{U}, \mathcal{W}\right)}{Z^*(G)} \tag{41}$$

Therefore, the conditional *dual partition function* of each instantiation $\mathcal{U},\mathcal{W}$ at $H(G_{nc}),D(G_{nc})$, stored in the table of the group, is defined by

$$\begin{aligned} & Z^*\left(G,H\left(G_{nc}\right),D\left(G_{nc}\right);\mathcal{U},\mathcal{W}\right) \\ & \quad = \prod_{r=1}^{h}\Bigg(Z^*\left(\kappa_r,\lambda_r;U_r,V_r\right) \\ & \qquad \times \exp\left(-\frac{E_{d5}\left(\kappa_r,\lambda_r,\alpha_r;U_r,V_r,A_r\right)+E_{d3}\left(\kappa_r,\lambda_r,\beta_r;U_r,V_r,B_r\right)}{RT}\right)\Bigg) \end{aligned} \tag{42}$$

Recall that the base pairs in $H(G_{nc})$ are adjacent. Therefore, due the constraints given by the overlapping positions within $D(G_{nc})$, and between $D(G_{nc})$ and $H(G_{nc})$, explained in "External loop" section, the number of possible instantiations $\mathcal{U},\mathcal{W}$ of $H(G_{nc}),D(G_{nc})$ is $\leq (6^h\cdot 4^{h+1})$.

In a similar way, sampling from the *closing group* $G_c$ closed by the base pair $(i,j)$, with $h+1$ base pairs in $H(G_c)$ and their corresponding flanking positions in $D(G_c)$ requires us to store the conditional *dual partition function* of each instantiation of nucleotides $\{X,Y,\mathcal{U},\mathcal{W}\}$ respectively at $i,j,H(G_c),D(G_c)$, where $\mathcal{U}=\{U_1,V_1,\ldots,U_h,V_h\}$ denotes an instantiation of the $h$ first base pairs $[(\kappa_1,\lambda_1),\ldots,(\kappa_h,\lambda_h)]$ in $H(G_c)$, $\mathcal{W}=\{A_1,B_2,\ldots,A_{h+1},B_{h+1}\}$ denotes an instantiation of the $2\cdot(h+1)$ flanking positions in $D(G_c)=[\alpha_1,\beta_1,\ldots,\alpha_{h+1},\beta_{h+1}]$, and $X,Y$ denotes an instantiation of $(i,j)$. The probability of the instantiation $\mathcal{U},\mathcal{W}$, given the nucleotides $X,Y$ is

$$\begin{aligned} & P\left(H(G_c)=\mathcal{U},D(G_c)=\mathcal{W}|i=X,j=Y\right) \\ & \quad =\frac{Z^*\left(G_c,i,j,H(G_c),D(G_c);X,Y,\mathcal{U},\mathcal{W}\right)}{Z^*\left(G_c;X,Y\right)} \end{aligned} \tag{43}$$

Then, the values stored in the table of the closing group correspond to the conditional *dual partition function* of each instantiation $\{X,Y,\mathcal{U},\mathcal{W}\}$ are given by $Z^*(G_c,i,j,H(G_c),D(G_c);X,Y,\mathcal{U},\mathcal{W})$, which is defined by the following expression:

$$\begin{aligned} & e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)}\cdot\exp\left(-\frac{e_{AU}(i,j,X,Y)}{RT}\right)\cdot\prod_{r=1}^{h}\Bigg(\Bigg(Z^*\left(\kappa_r,\lambda_r;U_r,V_r\right) \\ & \cdot\exp\left(-\frac{E_{d5}\left(\kappa_r,\lambda_r,\alpha_r;U_r,V_r,A_r\right)+E_{d3}\left(\kappa_r,\lambda_r,\beta_r;U_r,V_r,B_r\right)}{RT}\right)\Bigg)\Bigg) \\ & \cdot\exp\left(-\frac{E_{d3}\left(j,i,\alpha_{h+1};Y,X,A_{h+1}\right)+E_{d5}\left(j,i,\beta_{h+1};Y,X,B_{h+1}\right)}{RT}\right) \end{aligned} \tag{44}$$

As a final remark, we would like to recall that all the conditional *dual partition function* values are computed and stored in the forward step at the same time as the *dual partition function*. Therefore, despite the consequent increase of space complexity in the algorithm, the computation of the values required for correct sampling does not involve a greater time complexity.

## Scaling

The sequence partition function $Z^*(s_0)$ grows much faster than the usual structure partition function $Z(\mathbf{a})$, and so *scaling* must be used in the implementation. Let $C>2$ be a user-defined constant. By a slight modification of the previous recursions, we actually compute $Z^\dagger(i,j;X,Y)=\frac{Z^*(i,j;X,Y)}{C^{j-i+1}}$, and hence $Z^\dagger(s_0)=\frac{Z^*(s_0)}{C^n}$, where $n$ is the length of $s_0$. For instance, the analogue of Eq. (16) is

$$\begin{aligned} Z^\dagger &= \frac{Z^*\left(i,j;x,y\right)}{C^{j-i+1}} \\ &= e^{\left(-\frac{e^I_{AU}(x,y)}{RT}\right)}\cdot\frac{\exp\left(-\frac{H(j-i-1)}{RT}\right)}{C^{j-i+1}} \\ &\quad\times\left(\sum_{n_1,n_2\in\mathcal{N}}\exp\left(-\frac{mismatch(x,y,n_1,n_2)}{RT}\right)\cdot 4^{j-i-3}\right) \end{aligned} \tag{45}$$

and the analogue of Eq. (17) is

$$\begin{aligned} Z^\dagger &= \frac{Z^*(i,j;X,Y)}{C^{j-i+1}} \\ &= e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)}\cdot\frac{1}{2}\cdot\sum_{UV\in\mathcal{B}}\exp\left(-\frac{stack(X,Y,U,V)}{RT}\right) \\ &\quad\times Z^\dagger\left(i+1,j-1,U,V\right) \end{aligned} \tag{46}$$

This modification does not affect properties of sequences sampled from the low energy ensemble, since the same scaling factor appears in both the numerator and denominator of all conditional probabilities. For instance, the analogue of Eq. (31) is

$$\begin{aligned} & P\left(i+1=U,j-1=V|i=X,j=Y\right) \\ & \quad =\frac{Z^*\left(i,j,i+1,j-1;X,Y,U,V\right)}{Z^*\left(i,j;X,Y\right)} \\ & \quad =\frac{Z^\dagger\left(i,j,i+1,j-1;X,Y,U,V\right)}{Z^\dagger(i,j;X,Y)} \end{aligned} \tag{47}$$

## Controlling GC-content

The GC-content of an RNA sequence $\mathbf{a}=s_1,\ldots,s_n$ is the number of nucleotides that are either G or C. Instead of computing $Z^*(i,j;X,Y)$ and $Z^*(s_0)$, we can compute $Z^*(i,j;X,Y;\alpha)$ and $Z^*(s_0,\alpha)$, defined to be the corresponding partition *dual partition functions*, restricted to sequences having GC-content of $\alpha$. Note well that GC-content $\alpha$ includes the closing nucleotides $X$ and $Y$ respectively located at positions $i$ and $j$; i.e.

$$Z^*\left(i,j;X,Y;\alpha\right) = \sum_{\substack{a_i,\ldots,a_j, GC(a_i,\ldots,a_j)=\alpha \\ a_i=X, a_j=Y, a_{i+1},\ldots,a_{j-1}\in\mathcal{N}}} \exp\left(-E\left(a_i,\ldots,a_j;s_0[\,i,j]\right)/RT\right) \quad (48)$$

where $s_0[\,i,j]$ denotes the restriction of target structure $s_0$ to the interval $[\,i,j]$. We describe two particular subcases, to provide the idea of how modifications need to be undertaken.

### *Triloop*

Note that the number of RNA *sequences* of length $m$ having GC-content of $\alpha$ is $\binom{m}{\alpha}\cdot 2^{\alpha}\cdot 2^{m-\alpha} = \binom{m}{\alpha}\cdot 2^m \leq 4^m$, since $\alpha$ selected positions must be either G or C, yielding the term $2^{\alpha}$, while the remaining $m-\alpha$ positions must be either A or U, yielding the term $2^{m-\alpha}$. Assume that $\gamma(XY) = |\{X,Y\}\cap\{G,C\}| = \beta$. Then

$$Z^*\left(i,j;X,Y;\alpha\right) = e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)}\cdot\exp\left(-\frac{H(j-i-1)+e_{AU}(xy)}{RT}\right) \times\left(\binom{j-i-1}{(\alpha-\beta)}\cdot 2^{j-i-1} - |TriLoop_{x,y}| + \sum_{\substack{abc\in TriLoop_{x,y}\\ \gamma(abc)=\alpha-\beta}} \exp\left(-\frac{triloopE(xabcy)}{RT}\right)\right) \quad (49)$$

### *Multiloop and external loop*

Assume that $(i,j)$ closes a multiloop, which is a $(k+1)$-way junction with $\ell$ unpaired nucleotides. Assume that the ordered multiset of potential dangle positions is $D\ =[\,a_1,b_1,\ldots,a_{k+1},b_{k+1}]$, where $a_r = i_r - 1$ and $b_r = j_r + 1$ for $r = 1,\ldots,k$, and $a_{k+1} = i$ and $b_{k+1} = j$, and assume that there are $m$ unpaired positions that are not adjacent to a base pair in the multiloop. If $\mathbf{r}$ denotes an RNA sequence of arbitrary length, then let the function $\gamma(\mathbf{r})$ denote the GC-count in $\mathbf{r}$. Given an assignment of nucleotide base pairs $U_1V_1,\ldots,U_kV_k$ to $(i_1,j_1),\ldots,(i_k,j_k)$, where $U_rV_r \in \{GC,CG,AU,UA,GU,UG\}$, and given an assignment $A_1,B_1,\ldots,A_k,B_k$ of dangle nucleotides, where $A_r,B_r\in\mathcal{N}$, for $r=1,\ldots,k$, we let

$$\gamma(\mathbf{AB}) = \gamma\left(A_1,\ldots,A_k,B_1,\ldots,B_k\right). \quad (50)$$

Then the *dual partition function* of a multiloop with a GC-content of $\alpha$ is defined by setting $Z^*(i,j;X,Y;\alpha)$ equal to the following:

$$e^{\left(-\frac{e^I_{AU}(X,Y)}{RT}\right)}\cdot \sum_{\alpha_1+\cdots+\alpha_k\leq\alpha}\ \sum_{\{U_r,V_r\in\mathcal{B}:r=1,\ldots,k\}}\ \sum_{\{A_1,B_1,\ldots,A_k,B_k\in\mathcal{N}^{2k}\}} \exp\left(-\frac{a+b\cdot(k+1)+c\ell}{RT}\right)\cdot\left(\begin{matrix}(\ell-m)\\ \left(\alpha-\sum_{r=1}^k\alpha_r-\gamma(\mathbf{AB})\right)\end{matrix}\right)\cdot 2^{\ell} \times\exp\left(-\frac{e_{AU}(i,j,X,Y)}{RT}\right)\cdot\prod_{r=1}^{k}\Bigg(Z^*\left(i_r,j_r;U_r,V_r;\alpha_r\right) \times\exp\left(-\frac{E_{d5}\left(i_r,j_r,a_r;U_r,V_r,A_r\right)+E_{d3}\left(i_r,j_r,b_r;U_r,V_r,B_r\right)}{RT}\right) \times\exp\Bigg(-\frac{E_{d3}\left(j,i,a_{k+1};Y,X,A_{k+1}\right)+E_{d5}\left(j,i,b_{k+1};Y,X,B_{k+1}\right)}{RT} \quad (51)$$

Since the modification required in the remaining cases follows similar reasoning as in the treatment of the hairpin and external loop just described, the details for these remaining cases are not given.

An additional challenge of computing the *dual partition function* with GC-content control is the combinatorial problem of efficiently counting the number $N$ of instantiations of the external loop, consisting of all positions external to every base pair, with GC-content $k$, where the user can stipulate that certain positions are constrained to contain nucleotides consistent with IUPAC codes. To this end, we implemented the combinatorial algorithm defined in Supplementary Information.

### *Sampling with GC-content*

The implementation of sampling with GC-content is performed in a similar manner as described in "Sampling" section, with some notable differences.

First, the sampling function is redefined by *sample*$(k,T,i,j,X,Y,\alpha)$, where $k$ indicates the *base pair index* in the ordering defined by Eq. (10) for the base pair $(i,j)$ that is already instantiated by nucleotide pair $XY$, and $T$ designates the type of structural element closed by base pair $(i,j)$ in the target RNA secondary structure, as shown in Table 1. The function *sample*$(k,T,i,j,X,Y,\alpha)$ instantiates all positions of the loop having outer closing base pair $(i,j)$, including its inner closing base pair(s) and which returns the GC-content of the sampled loop. Moreover, the GC-content of the subsequence $\mathbf{a}[\,i+1,j-1] = (a_{i+1},\ldots,a_{j-1})$ will be $\alpha$ once the entire sequence $a_1,\ldots,a_n$ is sampled.

Second, `RNAdualPF` stores a conditional *dual partition function* table for each base pair $(i,j)$ and GC-content 0 to j-i-1. The function *sample*$(k,T,i,j,X,Y,\alpha)$ samples from the conditional *dual partition function* of those sequences which have exactly $\alpha$ Gs and Cs strictly between the positions $i$ and $j$, thus guaranteeing a GC-content of $\alpha$ for the subsequence $\mathbf{a}[i+1,j-1]$ once the entire sequence $a_1,\ldots,a_n$ is sampled. Note that $sample(k,T,i,j,X,Y,\alpha)$ samples only the loop closed by the already instantiated outer base pair $(i,j)$, and that $\alpha$ is the GC-content of the entire subsequence $\mathbf{a}[i+1,j+1] = a_{i+1},\ldots,a_{j-1}$ once the algorithm terminates. Only in the case that base pair $(i,j)$ closes a hairpin loop will it generally happen that the GC-content of the loop closed by $(i,j)$ is equal to $\alpha$.

Let $\alpha$ be the user-designated GC-content of sequences $\mathbf{a} = a_1,\ldots,a_n$ to be sampled from a target secondary structure having $\ell$ base pairs. The following pseudocode describes how to sample sequences $\mathbf{a} = a_1,\ldots,a_n$, whose GC-content exactly equals $\alpha$. Here, an external loop with $m$ components means that there are $m$ exterior base pairs $(i_1,j_1),\ldots,(i_m,j_m)$ such that all positions exterior to these base pairs are unpaired; i.e. each position $r \in \{1,\ldots,n\} - \cup_{c=1}^{m}\{i_c,\ldots,j_c\}$ is unpaired in the target structure.

**Algorithm 1** Sampling with user-specified GC-content $\alpha$

```
if external loop has $m$ components
  sample $\alpha_1,\ldots,\alpha_m,\beta$ with the following properties
    (a) $\alpha_1 + \ldots + \alpha_m + \beta = \alpha$
    (b) for $c=1$ to $m$, component $c$ has GC-content $\alpha_c$
    (c) GC-content of the external loop is $\beta$
    (d) sample the external loop
for $k=\ell$ down to 0
  let $(i,j)$ denote base pair with index $k$ and type $T$
  if $(i,j)$ is an exterior base pair closing the $c$th component
    //sample base pair with nucleotide pair $XY$ using roulette wheel
    $\alpha = \alpha_c$ //$\alpha$ now denotes GC-content of $c$th component
    z = random(0,1); cumProb = 0
    for XY in $\{AU, UA, GC, CG, GU, UG\}$
      $p = \frac{Z^*(i,j;X,Y;\alpha)}{Z^*(i,j;\alpha)}$
      cumProb += $p$
      if $z <$ cumProb
        instantiate base pair $(i,j)$ by $XY$
        $\alpha = \alpha -$GCcontent(XY)
        $sample(k,T,i,j,X,Y,\alpha)$
        $\alpha = \alpha -$sampledGC //subtract GC-content of sampled loop
        break //exit the innermost for-loop
  else // base pair $(i,j)$ with index $k$ is not exterior, hence is instantiated
    let $XY$ denote the nucleotides that instantiate $(i,j)$
    //sample loop and inner closing base pairs
    $sample(k,T,i,j,X,Y,\alpha)$
```

To clarify how the GC-content is sampled in a statistically rigorous manner, suppose that the user has specified the GC-content to be $\alpha$, and that $L$ is the external loop of the target structure $s_0$ having $m$ components, where the $c$th component has external closing base pair $(i_c,j_c)$. In computing the dual partition function, for all possible choices of non-negative integers $\alpha_1,\ldots,\alpha_m,\beta$ that sum to $\alpha$ and all $6^m$ possible assignments of Watson-Crick or wobble nucleotide pairs $X_1,Y_1,\ldots,X_m,Y_m$ to the base pairs $(i_1,j_1),\ldots,(i_m,j_m)$, the software RNAdualPF has computed the sum of $\sum_{c=1}^{m} Z_c^*(i_c,j_c;X_c,Y_c;\alpha_c)$ plus the Boltzmann factor of the external loop with GC-content $\beta$. Since the dual partition function $Z^*(s_0;\alpha)$ is the sum, taken over all values of $\alpha_1,\ldots,\alpha_m,\beta$ and all Watson-Crick and wobble pair assignments to the external base pairs, RNAdualPF can then use the roulette wheel method to sample values $\alpha_1,\ldots,\alpha_m,\beta$ and $X_1,Y_1,\ldots,X_m,Y_m$ in a statistically rigorous manner. Multiloops, and other structural elements, which contain unpaired regions whose sequence does not contribute to the free energy of the structure, are handled in a analogous manner.

## Results

### Robustness and plasticity of *C. elegans* miRNAs and *E. coli* sncRNAs

In [1] Borenstein and Ruppin used version 1.4 of the Vienna RNA Package [7] to generate 1000 RNA sequences per wild type precursor microRNA (pre-miRNA) extracted from the database Rfam 1.0 [19], with the property that each of the 1000 control sequences folded into the wild type pre-miRNA structure – i.e. the minimum free energy (MFE) structure of each of the 1000 control sequences was identical to the MFE structure of the wild type pre-miRNA. Based on these computational experiments, Borenstein and Ruppin asserted that the "structure of miRNA precursor stem–loops exhibits a significantly high level of mutational robustness in comparison with random RNA sequences with similar stem–loop structures". Noting that the Vienna RNA Package inverse folding program RNAinverse does not control for GC-content or other sequence compositional bias, the authors performed a second computational experiment, in which control sequences not only folded into the target wild type structure, but also had similar dinucleotide composition to that of wild type pre-miRNA (Jensen-Shannon divergence less than 0.01). Since the filtering step required enormous run time and computational resources, the authors restricted their attention to a small set of 211 microRNAs, and generated only 100 control sequences per microRNA – note here that RNAinverse cannot control for GC-content. Borenstein and Ruppin concluded that robustness of precursor microRNAs was not the byproduct of a base composition bias or of thermodynamic stability.

Subsequently Rodrigo et al. [3] undertook a similar analysis for bacterial small noncoding RNAs (sncRNA), also using the program `RNAinverse`, albeit using somewhat different definitions – see precise definitions in "Formal definitions of robustness" section. The main finding of [3] was that bacterial sncRNAs are *not significantly robust* when compared with 1000 sequences having the same structure, as computed by `RNAinverse`; however, the authors found that bacterial sncRNAs tend to be significantly *plastic*, in the sense that the ensemble of low energy structures are structurally diverse. Unlike the case of precursor microRNAs [1], Rodrigo et al. did not control for sequence compositional bias.

Using `RNAdualPF`, we performed similar computational experiments on 250 precursor microRNAs of *C. elegans* from miRBase 20 [11] and for the bacterial small noncoding RNAs of [3]. Below, we discuss each case separately.

For each *C. elegans* pre-miRNA, we used `RNAdualPF` to sample 2000 sequences with no control over GC-content and 2000 sequences whose GC-content was identical with that of the wild type pre-miRNA. Moreover, each control sequence *approximately* folded into the MFE wild type pre-miRNA structure as computed by Vienna RNA Package 2.1.9 [2]. Table 2 shows that length-normalized base pair distance between the MFE structure of the control sequence and that of the pre-miRNA is on average 0.09 ± 0.04 for default use of `RNAdualPF` with control over GC-content, and 0.06 ± 0.03 when GC-content of each control sequence is identical to that of the corresponding wild type pre-miRNA. Additional measures in Table 2 show that sequences sampled from `RNAdualPF` (1) are only modestly more stable thermodynamically, (2) the ensemble of low energy structures of control sequences deviate slightly more from the target pre-miRNA structure, as is the case for wild type pre-miRNA sequences, as mesured by ensemble defect [20], expected base pair distance to target [9], expected proportion of native contacts (called ensemble neutrality in [21]), average positional entropy [22], Morgan-Higgs structural diversity [23], and Vienna structural diversity (called *ensemble diversity* in [14]).

Tables 3 and 4 display a similar analysis of the collection of bacterial small noncoding RNAs of [3] and of Rfam 12.0 database [24]. For the Rfam database, we selected one sequence from each of the ≈ 2500 Rfam families, with the property that the MFE structure of the sequence most resembled the Rfam consensus structure – i.e. whose MFE structure has smallest base pair distance to the consensus structure. These tables show similar trends

**Table 2** Analysis of *C. elegans* precursor microRNA from the database miRBase 20 [11]

| MEASURE | Def. Exact GC | Def. No GC | MFE Exact GC | MFE No GC | WT |
|---|---|---|---|---|---|
| BP DIST TARGET | 0.06±0.03 | 0.09±0.04 | 0±0 | 0±0 | 0±0 |
| ENERGY MFE | –0.53±0.11 | –0.85±0.12 | –0.48±0.12 | –0.79±0.14 | –0.38±0.11 |
| ENERGY TARGET | –0.46±0.12 | –0.78±0.14 | –0.48±0.12 | –0.79±0.14 | –0.38±0.11 |
| ENSEMBLE DEFECT | 0.12±0.04 | 0.14±0.06 | 0.05±0.02 | 0.05±0.02 | 0.08±0.05 |
| EXP BP DIST | 0.07±0.03 | 0.1±0.04 | 0.03±0.01 | 0.03±0.01 | 0.05±0.03 |
| PROP NAT CONTACT | 0.93±0.04 | 0.9±0.06 | 0.96±0.02 | 0.96±0.02 | 0.92±0.05 |
| POS ENTROPY | 0.14±0.05 | 0.14±0.05 | 0.13±0.05 | 0.12±0.05 | 0.2±0.11 |
| GC CONTENT | 42.88±9.14 | 82.07±3.5 | 42.9±9.15 | 80.92±4.09 | 42.88±9.14 |
| LN DUAL PROB | –95.4±21.03 | –51.43±11.98 | –102.73±22.81 | –59.52±14.64 | –117.11±25.94 |
| LN PROB | -10.81±4.05 | –10.95±4.68 | –1.38±0.55 | –0.96±0.45 | –2.02±0.97 |
| MH STR DIV | 0.08±0.03 | 0.08±0.03 | 0.07±0.03 | 0.06±0.03 | 0.11±0.06 |
| VIENNA STR DIV | 0.05±0.02 | 0.05±0.02 | 0.04±0.02 | 0.04±0.02 | 0.07±0.04 |

For each of the 500 wild type (WT) pre-miRNA sequences, `RNAdualPF` sampled sequences, either having exactly the same GC-content as the WT sequence ('Exact GC') or with no control over GC-content ('No GC'). The designation 'MFE' indicates that the sampled sequences were subsequently filtered to retain only those, whose minimum free energy structure is identical to the MFE structure of the corresponding WT pre-miRNA; otherwise, the designation 'Def' is used to indicate the default output of `RNAdualPF`, without the subsequent filtering step. For each WT pre-miRNA sequence, `RNAdualPF` generated 2000 sequences for the default case *Def* (no subsequent filtering), and 500 sequences for the non-default case *MFE*, such that sample MFE structure is identical to WT MFE structure. Various measures were used to compare the properties of `RNAdualPF` sampled sequences to those of wild type sequences: *BP DIST TARGET*: length-normalized average base pair distance $d_{BP}(s_0, s^*)$ between the MFE structure $s_0$ of sequences sampled by `RNAdualPF` and the target structure $s^*$. *ENERGY MFE*: length-normalized average free energy $E(s_0)$ of MFE structure $s_0$. *ENERGY TARGET*: length-normalized average free energy $E(s^*)$ of target $s^*$ for the respective sequences. *ENSEMBLE DEFECT*: length-normalized expected Hamming distance to target $s^*$ [20]. *EXP BP DIST*: length-normalized expected base pair distance to target $s^*$ [9]. *PROP NAT CONTACT*: expected proportion of base pairs of target $s^*$ that occur in the MFE structure, i.e. $\langle \frac{|s_0 \cap s^*|}{|s^*|} \rangle$. *POS ENTROPY*: average positional entropy [22]. *GC CONTENT*: average proportion of positions occupied by G or C. *LN DUAL PROB*: average natural logarithm of the dual probability $\exp(-E(\mathbf{a}, s)/RT)/Z * (\mathbf{s})$ that sequence **a** adopts the structure *s*. *LN PROB*: average natural logarithm of the probability $\exp(-E(\mathbf{a}, s)/RT)/Z(\mathbf{a})$ that sequence **a** adopts the structure *s*. *MH STR DIV*: length-normalized Morgan-Higgs structural diversity [23]. *VIENNA STR DIV*: length-normalized Vienna structural diversity, called *ensemble diversity* in [14]. Values of all measures for default sampled sequences having GC-content within 5 % of wild type GC-content (not shown) are essential identical to those of exact GC-content control

**Table 3** Analysis of bacterial RNAs [3]

| MEASURE | GC 5 % | Exact GC | No GC | WT |
|---|---|---|---|---|
| BP DIST TARGET | 0.08±0.04 | 0.08±0.04 | 0.14±0.06 | 0±0 |
| ENERGY MFE | −0.44±0.1 | −0.44±0.1 | −0.63±0.14 | −0.29±0.1 |
| ENERGY TARGET | −0.39±0.12 | −0.39±0.12 | −0.54±0.16 | −0.29±0.1 |
| ENSEMBLE DEFECT | 0.16±0.08 | 0.16±0.08 | 0.23±0.1 | 0.14±0.09 |
| EXP BP DIST | 0.09±0.04 | 0.09±0.04 | 0.15±0.06 | 0.09±0.06 |
| PROP NAT CONTACT | 0.89±0.09 | 0.89±0.09 | 0.8±0.12 | 0.83±0.15 |
| POS ENTROPY | 0.19±0.08 | 0.19±0.08 | 0.23±0.09 | 0.35±0.18 |
| GC CONTENT | 48.33±7.02 | 48.34±7.02 | 74.75±5.55 | 48.34±7.02 |
| LN DUAL PROB | −94.59±27.22 | −94.54±27.19 | −66.37±16.96 | −117.22±33.37 |
| LN PROB | −10.12±4.04 | −10.09±4.03 | −13.71±5.16 | −2.34±0.95 |
| MH STR DIV | 0.1±0.04 | 0.1±0.04 | 0.13±0.05 | 0.18±0.09 |
| VIENNA STR DIV | 0.06±0.03 | 0.06±0.03 | 0.09±0.03 | 0.11±0.06 |

See Table 2 for an explanation of column headers and various measures. Since bacterial noncoding RNA is generally much longer than precursor microRNA, no subsequent filtering step was undertaken to ensure that sample sequence MFE structure is identical to that of wild type pre-miRNA. However an additional column is given for sequences required by `RNAdualPF` to have GC-content is within 5 % of WT value. (column header GC 5 %)

as those displayed in Table 2, although values are larger due to increased sequence length of bacterial sncRNA and sequences from Rfam.

In agreement with [1], the left panel of Fig. 3 shows that *C. elegans* miRNA is significantly robust (Z-score of 0.61 ± 1.55, 2-tailed T-test *p*-value $2.2 \times 10^{-9}$), *provided* that GC-content is *not* controlled. However, in contrast to [1], when GC-content is controlled, we find that *C. elegans* miRNA is significantly *non-robust* (Z-score of $-1.3 \pm 2.9$, 2-tailed T-test *p*-value $1.5 \times 10^{-11}$). To corroborate our findings, for each wild type *C. elegans* pre-miRNA, we performed a second computational experiment, to generate 500 sequences with no control over GC-content and 500 sequences whose GC-content was identical with that of the wild type pre-miRNA. In contrast to the first experiment, we used `RNAdualPF` to generate sufficiently many sequences to subsequently select 500 sequences (no GC-control) and 500 sequences (GC-content equal to wild type pre-miRNA), each of whose MFE structure was identical to that of wild type pre-miRNA. The left panel of Fig. 4 shows that when GC-content is not controlled, *C. elegans* precursor microRNAs are statistically robust (Z-score 0.51 ± 1.44, *p*-value $7.3 \times 10^{-8}$), in agreement with the main result of Borenstein and Ruppin [1]. However, when GC-content of control sequences is identical to that of wild type precursor microRNA, we confirm that *C. elegans* pre-miRNA is *statistically non-robust* (Z-score $-1.23 \pm 2.78$, *p*-value

**Table 4** Analysis of the Rfam 12.0 database

| MEASURE | GC 5 % | Exact GC | No GC | WT |
|---|---|---|---|---|
| BP DIST TARGET | 0.1±0.05 | 0.1±0.05 | 0.16±0.07 | 0±0 |
| ENERGY MFE | −0.43±0.13 | −0.43±0.13 | −0.64±0.16 | −0.28±0.13 |
| ENERGY TARGET | −0.36±0.14 | −0.36±0.14 | −0.54±0.19 | −0.28±0.13 |
| ENSEMBLE DEFECT | 0.18±0.07 | 0.18±0.07 | 0.25±0.11 | 0.16±0.12 |
| EXP BP DIST | 0.11±0.05 | 0.11±0.05 | 0.17±0.07 | 0.1±0.08 |
| PROP NAT CONTACT | 0.87±0.09 | 0.87±0.09 | 0.78±0.14 | 0.81±0.17 |
| POS ENTROPY | 0.22±0.09 | 0.21±0.09 | 0.25±0.1 | 0.4±0.25 |
| GC CONTENT | 46.27±10.91 | 46.27±10.91 | 75.12±5.89 | 46.27±10.91 |
| LN DUAL PROB | −110.35±54.12 | −110.34±54.12 | −73.5±33.32 | −136.7±65.62 |
| LN PROB | −13.94±7.74 | −13.9±7.71 | −18.11±10.59 | −2.83±1.71 |
| MH STR DIV | 0.12±0.05 | 0.12±0.05 | 0.13±0.05 | 0.2±0.12 |
| VIENNA STR DIV | 0.07±0.03 | 0.07±0.03 | 0.09±0.04 | 0.13±0.08 |

For each RNA family from Rfam 12.0, we selected that sequence whose MFE structure had smallest base pair distance to the Rfam consensus structure for the family. These sequences constituted the collection WT. See Table 3 for an explanation of column headers and various measures

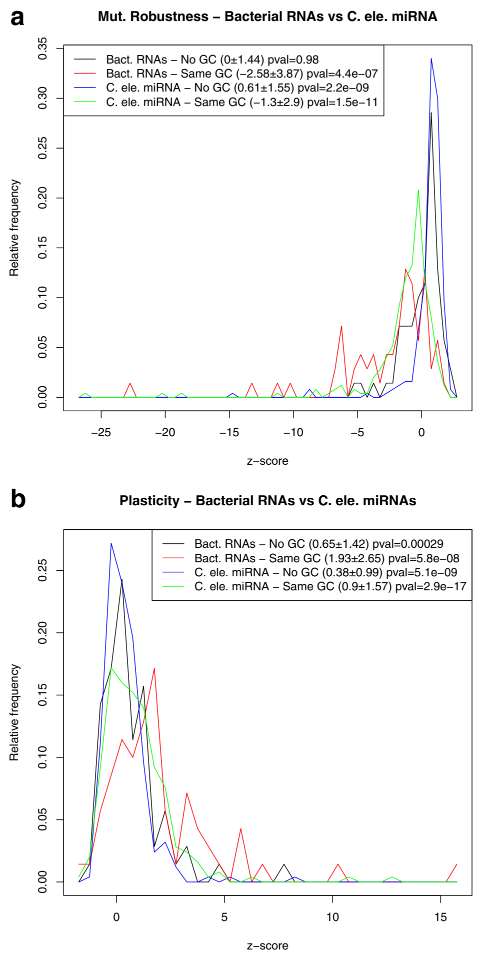

**Fig. 3** Z-scores of *mutational robustness* (**a**) and of *plasticity* (**b**) are presented for the bacterial small noncoding RNA (sncRNA) collection from [3] and for *C. elegans* precursor microRNA (pre-miRNA) from miRBase 20. For each wild type pre-miRNA [resp. sncRNA] wild type sequence, RNAdualPF sampled 2000 [resp. 1000] sequences using the minimum free energy structure of the wild type sequence as target structure. The GC-content of the sampled sequences was either required to be exactly that of the wild type sequence, or not (default mode of RNAdualPF), as indicated in the legend. Sampled sequences were used to compute the mutational robustness and plasticity, as explained in the main text. Note that *C. elegans* miRNA is significantly *robust* if GC-content is not controlled, but significantly *non-robust* if GC-content of RNAdualPF samples is identical to that of wild type pre-miRNA. Similarly, bacterial sncRNAs are not significantly robust if GC-content is not controlled, but significantly non-robust when GC-content is identical to that of wild type sncRNA. For this figure, mutational robustness of RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ is defined by $1 - \frac{\langle D_{\mathrm{BP}} \rangle}{n}$, where ensemble distance $D_{\mathrm{bp}}(\mathbf{a}, \mathbf{b})$ between two length $n$ sequences **a** and **b** is defined in [14], and the average ensemble distance from all single-point mutants of **a** is defined by $\langle D_{\mathrm{BP}} \rangle = \sum_{\mathbf{b}} \frac{D_{\mathrm{BP}}(\mathbf{a},\mathbf{b})}{3n}$ where the sum is taken over all single-point mutants **b** of **a**. We use this notation of mutational robustness, rather than the notion defined in [3], since the latter notion is not a true metric, as explained in "Formal definitions of robustness" section. The plasticity $P = \frac{\langle D_{\mathrm{V}} \rangle}{n/2} = \sum_{i<j} \frac{p_{ij}(1-p_{ij})}{n}$ is defined in [3] as normalized *ensemble diversity*, where ensemble diversity [14] (Vienna structural diversity) $D_{\mathrm{V}}$ is defined by Eq. (4)

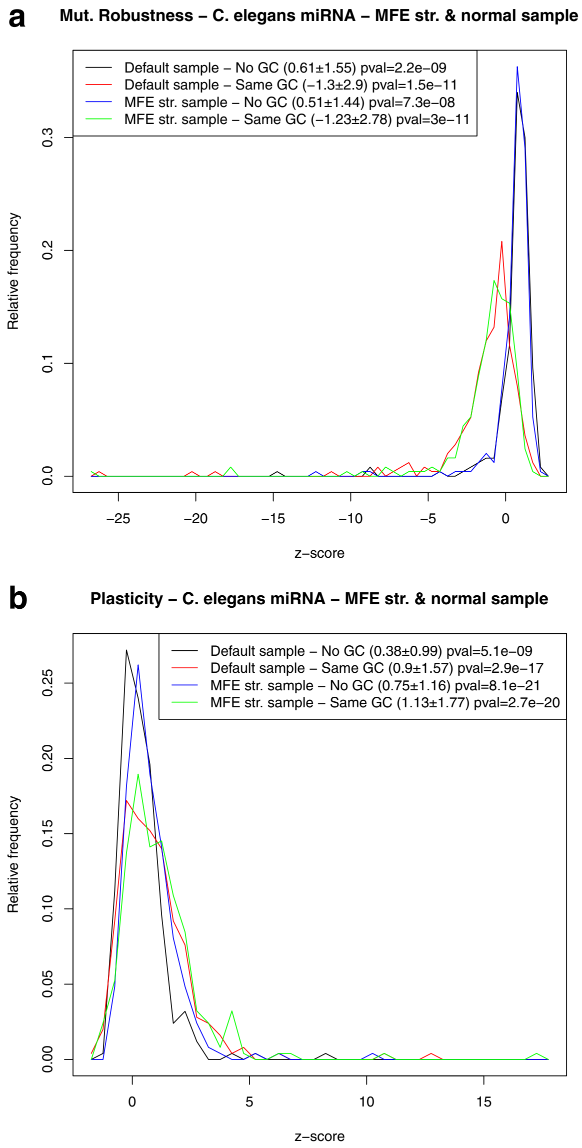

**Fig. 4** Z-scores of *mutational robustness* (**a**) and of *plasticity* (**b**) are presented for *C. elegans* microRNA from Rfam 12.0. Robustness and plasticity were measured as explained in Fig. 3. For each wild type (WT) sequence, 2000 sequences were sampled from RNAdualPF both with and without control over GC-content (default sample), while 500 sequences were generated by RNAdualPF, both with and without control over GC-content, to *exactly* fold into the target structure (MFE str. sample). This was achieved by repeatedly sampling sequences in order to obtain a sequence, whose MFE structure was identical to the target structure. The number of sequences necessary to sample in order to obtain one sequence that folds into the target structure was variable, depending on the target structure and GC-content – in many cases, only 10 samples were necessary per selected sequence, in some cases 200 samples were necessary. and in one specific case 5000 samples were required. By *control* over GC-content, we mean that all sampled sequences have identical GC-content to the wild type sequence

$3 \times 10^{-11}$). Note that our finding, which is in opposition to results of Borenstein and Ruppin [1], is based on a larger data set of precursor microRNAs, each of which has a larger control set, than in the analysis of [1].

Turning now to the analysis of bacterial small noncoding RNAs, we find that sncRNAs are not significantly robust (Z-score $0.0 \pm 1.4$, $p$-value 0.98) when GC-content is *not controlled*, confirming a result from Rodrigo et al. [3]. However, when GC-content of the sequences sampled from RNAdualPF is required to be identical to that of wild type sncRNA, bacterial sncRNAs are see to be *significantly non-robust* (Z-score $-2.58 \pm 3.87$, $p$-value of $4.4 \times 10^{-7}$). Note here that Rodrigo et al. used RNAinverse in their computational experiments, hence could not consider the case with control over GC-content. The left panels of Figs. 3a and 4 summarize our findings that precursor microRNAs [resp. bacterial sncRNAs] are significantly non-robust [resp. not significantly robust] with respect to a control set of 2000 [resp. 1000] sequences generated by RNAdualPF with identical GC-content to that of the wild type sequence.

Finally, in our analysis of *plasticity*, the right panels of Figs. 3 and 4 show that both *C. elegans* and bacterial small noncoding RNAs exhibit more plasticity when compared with control sequences for which GC-content is not controlled, as well as when compared with control sequences for which GC-content is identical to that of wild type sequences.

### Structural RNA has higher free energy than expected

In Figure 4 of [9], we showed that the free energy $E_0$ of the minimum free energy (MFE) structure $s_0$ of *E. coli* val-tRNA (accession RV1600 from Sprinzl database [25] tdbR00000454 from tRNAdb [26]), is much *higher* (less favorable) than the average free energy $\langle E \rangle$ of over four million RNAs having the same MFE structure $s_0$ as that of *E. coli* val-tRNA. Here, *E. coli* val-tRNA RV1600 was selected, because its MFE structure $s_0$ is identical to the Rfam consensus structure for tRNA family RF00005. This preliminary result suggests that naturally occurring transfer RNAs may be under selective pressure to be only marginally thermodynamically stable. Since it took a number of days for RNAiFold [4, 9] to return over four million solutions of the inverse folding problem for the tRNA target structure, we now describe how RNAdualPF can be used to compute the Boltzmann expected free energy of literally all sequences $a_1, \ldots, a_n$ with respect to an arbitrary target structure $s_0$. In this manner, we confirm our preliminary finding concerning *E. coli* val-tRNA, and show that the folding energy of structural RNA from the Rfam database is much higher (less favorable) than expected. Before presenting results, we need some definitions.

For the Turner nearest neighbor energy model [13], the free energy of a secondary structure $s$ of an RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ depends on the (absolute) temperature $T_0$. To indicate this dependence, we write $E(\mathbf{a}, s, T_0)$, where in the sequel, $T_0$ will be designated as *table temperature*, i.e. the temperature for which parameters from the Turner energy tables are applied. For an arbitrary, but fixed secondary structure $s_0$ of length $n$, the *dual partition function* at temperature $T_0$ is defined by

$$Z(s_0, T_0, T) = \sum_{\mathbf{a}} \exp\left(-E\left(\mathbf{a}, s_0, T_0\right)/RT\right) \tag{52}$$

where the sum is taken over all RNA sequences $\mathbf{a} = a_1, \ldots, a_n$ of length $n$. Note that $T_0$ indicates the (table) temperature at which the energy of a structure $s_0$ and nucleotide sequence $\mathbf{a}$ is evaluated using the Turner parameters, while all other occurrences of the temperature variable are designated by $T$, which we call *formal temperature*. The distinction between formal and table temperature is made to allow us to use finite difference approximations to derivatives with respect to the *formal temperature* when when we compute *dual expected energy* and *dual conformational entropy* below (see [27] for more explanation). When table temperature $T_0$ equals formal temperature $T$, and the temperature is clear from the context, we write $Z^*(s_0)$; if the target structure $s_0$ is also clear from the context, then we write $Z^*$. A similar remark applies to the other thermodynamic functions $p^*, G^*, \langle E^* \rangle, S^*$, which we now define.

The *dual Boltzmann probability* $p^*(\mathbf{a})$ is defined by

$$p^*\left(\mathbf{a}, s_0, T_0, T\right) = \frac{\exp\left(-E\left(\mathbf{a}, s_0, T_0\right)\right)}{Z^*\left(s_0, T_0, T\right)} \tag{53}$$

The *dual ensemble free energy* $G^*(s_0)$ is defined by

$$G^* = G^*(s_0) = G\left(s_0, T_0, T\right) = -RT \ln Z^*\left(s_0, T_0, T\right) \tag{54}$$

where $R \approx 1.987$ cal/(mol K) is the universal gas constant. The *dual expected (free) energy* $\langle E^*(s_0) \rangle$ is defined by

$$\left\langle E^*\left(s_0, T_0, T\right)\right\rangle = \sum_{\mathbf{a}} E\left(\mathbf{a}, s_0, T_0\right) \cdot p\left(\mathbf{a}, s_0, T_0, T\right) \tag{55}$$

Straightforward derivations analogous to those in [27] yield the following expressions for *dual expected energy* $\langle E^* \rangle$ and *dual entropy* $S^*$:

$$\left\langle E^*\left(s_0, T_0, T\right)\right\rangle = RT^2 \cdot \frac{\partial}{\partial T} \left(\ln Z^*(s_0, T_0, T)\right)_{T=T_0} \tag{56}$$

$$S^*\left(s_0, T_0, T\right) = \frac{\left\langle E\left(\mathbf{a}, s_0, T_0, T\right)\right\rangle - G^*\left(s_0, T_0, T\right)}{T} \tag{57}$$

Programs to compute *dual expected energy* $\langle E^* \rangle$, *dual conformational entropy* $S^*$, and *dual heat capacity* $C_p^*$ are provided at our web site. We do not elaborate further on dual entropy or dual heat capacity, since at the present time we have found no compelling applications.

Figure 5 shows that structural RNAs have *higher* free energy with respect to their native structure, hence are thermodynamically *less stable*, than expected, – even when expectations are taken over all sequences having the same GC-content as that of wild type sequences. We believe that this insight could be important when designing functional synthetic RNAs. To generate Fig. 5, we proceeded as follows. For each family from the Rfam 12.0 database [24], we took the family consensus structure $s_c$, and computed $\langle E(s_c)\rangle$. Additionally, for each Rfam family, we selected that sequence $\mathbf{a}_0$, whose minimum free energy (MFE) structure $s_0$ has smallest base pair distance to the consensus structure $s_c$. We computed the expected energy $\langle E(s_0)\rangle$, as well as the free energies $E(\mathbf{a}, s_c)$ and $E(\mathbf{a}, s_0)$. Figure 5 displays box-and-whiskers plots for the fold change $\frac{\langle E(s_c)\rangle}{E(\mathbf{a}_0,s_c)}$ for the consensus structure and the fold change $\frac{\langle E(s_0)\rangle}{E(\mathbf{a}_0,s_0)}$ for the minimum free energy structure. Since the dual Boltzmann probability $p^*(\mathbf{a}, s_0)$ is generally larger for sequences **a** having higher GC-content (as stacked base pairs involving GC,CG have lower free energy than those involving AU,UA,GU,UG), RNAdualPF computes as well the *dual partition function* for GC-content $k$, defined by

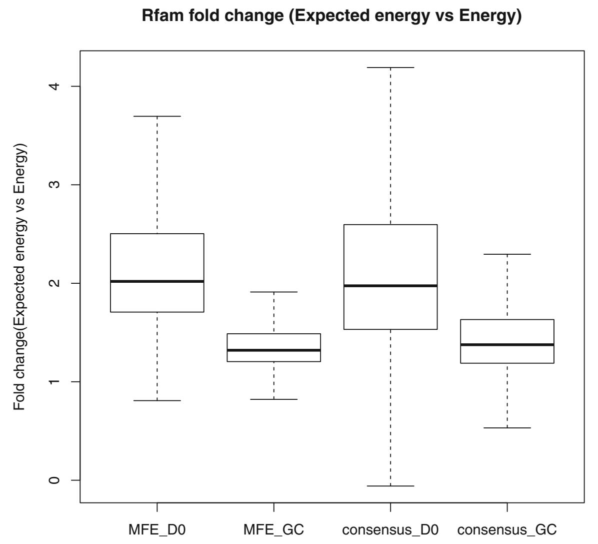

**Fig. 5** Analysis of expected free energy $\langle E\rangle$ for structures in Rfam 12.0 [24]. Given a secondary structure $s$, the expected free energy of all sequences **a** with respect to $s$ is defined by $\langle E(s)\rangle = \sum_{\mathbf{a}} E(\mathbf{a}, s) \cdot \frac{\exp(-E(\mathbf{a},s)/RT)}{Z^*(\mathbf{a},s)}$, where $Z^*$ is the *dual partition function* defined in equation (7). For each Rfam family, we took the family consensus structure $s_c$, and computed $\langle E(s_c)\rangle$. Additionally, for each Rfam family, we selected that sequence $\mathbf{a}_0$, whose minimum free energy (MFE) structure $s_0$ has smallest base pair distance to the consensus structure $s_c$. The expected energy $\langle E(s_0)\rangle$ was computed, as well as the free energies $E(\mathbf{a}, s_c)$ and $E(\mathbf{a}, s_0)$. The fold change $\frac{\langle E(s_c)\rangle}{E(\mathbf{a}_0,s_c)}$ for the consensus structure and the fold change $\frac{\langle E(s_0)\rangle}{E(\mathbf{a}_0,s_0)}$ for the minimum free energy structure were computed. The box-and-whiskers plots show the mean, 25th and 75th percentile, minimum and maximum values. As indicated in the legend, these computations were performed either with respect to all sequences or with respect to all sequences having the same (exact) GC-content. These data clearly indicate that natural RNA sequences, whose MFE structures most closely resemble the Rfam consensus structures, have *higher* free energy than expected

$$Z^*(s_0, k) = \sum_{\substack{\mathbf{a}\text{ such that}\\ \text{GC-content}=k}} \exp(-E(\mathbf{a}, s_0)/RT) \tag{58}$$

In this fashion, we can exactly compute the *dual expected energy* $\langle E^*(s_0, k)\rangle$ of all sequences having GC-content $k$ which approximately fold into target structure $s_0$. Tables 2, 3 and 4 analyze what we mean by *approximately* folding into the target structure – i.e. sequences **a** are preferentially sampled when free energy $E(\mathbf{a}, s_0)$ is low, hence have large dual Boltzmann probability. RNAdualPF, even when exact GC-content is controlled, is faster than inverse folding programs by orders of magnitude, hence providing an effective alternative manner of solving inverse folding.

## Conclusion

In this paper we describe the algorithm and software RNAdualPF, which computes the *dual partition function* $Z^*$, defined as the sum of Boltzmann factors $\exp(-E(\mathbf{a}, s_0)/RT)$ of all *sequences* **a** with respect to the target structure $s_0$. Using RNAdualPF, we efficiently sample RNA sequences that (approximately) fold into $s_0$, where additionally the user can specify IUPAC sequence constraints at certain positions, and whether to include dangles (energy terms for stacked, single-stranded nucleotides). Moreover, the user can require that all sampled sequences have a precisely specified GC-content, since, optionally, we compute the *dual partition function* $Z^*(k)$ simultaneously for all values $k = G + C$. This sampling strategy is complementary to the use of RNAiFold [4], since it allows the study of the properties of long RNA structures whose number of solutions for the inverse folding problem is astronomically large.

We use RNAdualPF to corroborate previous studies [1] using RNAinverse [2], by confirming that precursor microRNAs are significantly mutationally robust when GC-content is not controlled. However, in contrast to [1], we find that precursor microRNAs are significantly *non-robust* when GC-content is controlled. We confirm and extend previous findings [3] that bacterial small noncoding RNAs display plasticity (structural diversity) and are not statistically robust, when GC-content is not controlled. Additionally, we obtain the new finding that when when GC-content is controlled, bacterial small noncoding RNAs are significantly non-robust, as in the case of precursor microRNAs. One possible reason for the discrepancy between our results and those of [1] could be related with the fact that the energy parameters of Vienna RNA Package 1.4 (Turner 1999 parameters used in the computational experiments of [1]) differ from those of Vienna RNA Package 2.1.9 (Turner 2004 parameters used in the current study with `RNAdualPF`). Another possible reason is that the inverse folding solutions returned by the program `RNAinverse` used in [1] show a different bias than sequences returned by `RNAdualPF` (in this context, we mean the inverse folding solutions filtered from the sequences returned by `RNAdualPF`).

As mentioned in the Introduction, there is a relation between our C program `RNAdualPF` and the Python program `IncaRNAtion` [12], although our work is independent of that of Reinharz et al. [12]. `IncaRNAtion` is a weighted sampling algorithm that computes the dual partition function for a simple energy model, which only considers base stacking free energies – unlike `RNAdualPF`, the program `IncaRNAtion` includes no energy contributions for hairpins, bulges, internal loops, multiloops, dangles, or mismatches. If the user specifies a desired GC-content $\alpha$, then `IncaRNAtion` does not compute the dual partition function for GC-content, but rather applies an adjustable heuristic so that after a suitable *burn-in* period, sequences tend to approximately have GC-content $\alpha$. See Table 6.2 of [28] for benchmarking results on `RNAdualPF` and `IncaRNAtion`, which show conclusively that `RNAdualPF` is not only faster, but its sequences have a higher probability of folding into the target structure, its sequences have a smaller GC-content in default mode, where GC-content is not controlled, etc.

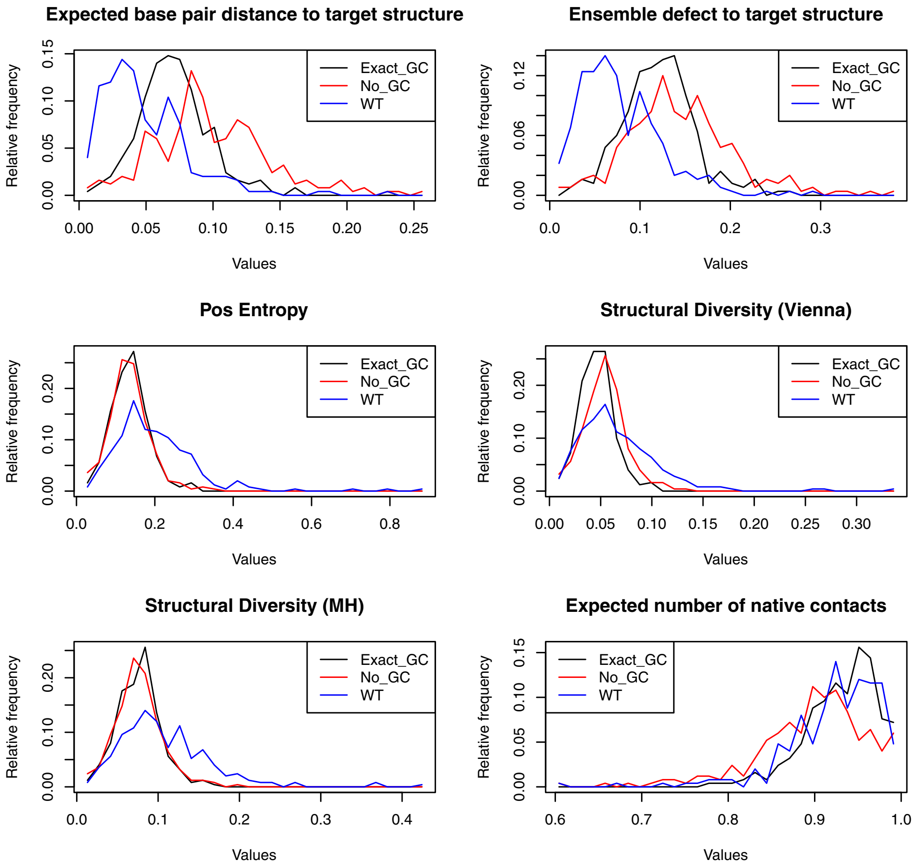

**Fig. 6** For each of the 250 *C. elegans* precursor microRNAs from miRBase 20 and for each of the following cases (**a**), indicated in *black*, and (**b**), indicated in *red*, `RNAdualPF` sampled 2000 sequences without any subsequent filtering step. Case (**a**) - *black* lines: All `RNAdualPF` sequences have GC-content exactly equal to that of the Rfam sequence (Exact GC). Case (**b**) - *red* lines: `RNAdualPF` was used in default mode, without controlling GC-content (No GC). Case (**c**) - *blue* lines: wild type (WT) *C. elegans* data. Density plots are shown for (1) the expected base pair distance to target structure $s_0$ [9], (2) the ensemble defect to target structure $s_0$ [20], (3) the positional entropy [22], (4) Vienna structural diversity (called ensemble diversity in [14]), (5) Morgan-Higgs diversity [23], (6) expected proportion of native contacts (called ensemble neutrality in [21]). All measures were normalized by sequence length

Our original motivation in designing RNAdualPF was to generate an *unbiased* sample of *near-solutions* (or by subsequent selection of *solutions*) to the inverse folding problem. At present, it seems clear that no program can claim to generate an unbiased sample of inverse folding solutions, since (1) the solution space so large that this hypothesis cannot be tested by brute force methods, and (2) different inverse folding algorithms return solution sequences having different properties, as shown in Table 2 of [4]. Nevertheless, in the same manner that structures sampled by the algorithm of Ding and Lawrence [16] constitute an unbiased, representative set of low energy secondary structures for a given RNA sequence, as implemented in Sfold [10] and RNAsubopt -p [2], the collection of RNA sequences sampled by the algorithm RNAdualPF constitute an unbiased, representative set of sequences having low energy with respect to a given target structure $s_0$. Although the minimum free energy structure of such sequences may indeed be distinct from $s_0$, it is likely that the MFE structure and the target $s_0$ be similar, as shown in Tables 2, 3 and 4. Moreover, Fig. 6 presents relative frequency plots that suggest that when GC-content is controlled, the sequences returned by RNAdualPF have similar properties to those of wild type sequences: (1) similar expected base pair distance to the wild type target structure [9], (2) similar ensemble defect to the target wild type structure [20], (3) similar positional entropy [22], (4) similar Vienna structural diversity (called ensemble diversity in [14]), (5) similar Morgan-Higgs diversity [23], (6) similar expected proportion of native contacts (called ensemble neutrality in [21]). These graphs were produced by using RNAdualPF to sample 2,000 sequences for each of the 250 *C. elegans* precursor microRNAs from the miRBase 20 database [11], in each of the following cases: (a) GC-content identical to that of the Rfam sequence, (b) no control for the GC-content. Figure 7 presents additional data, computed in the same manner for *C. elegans* pre-miRNA from

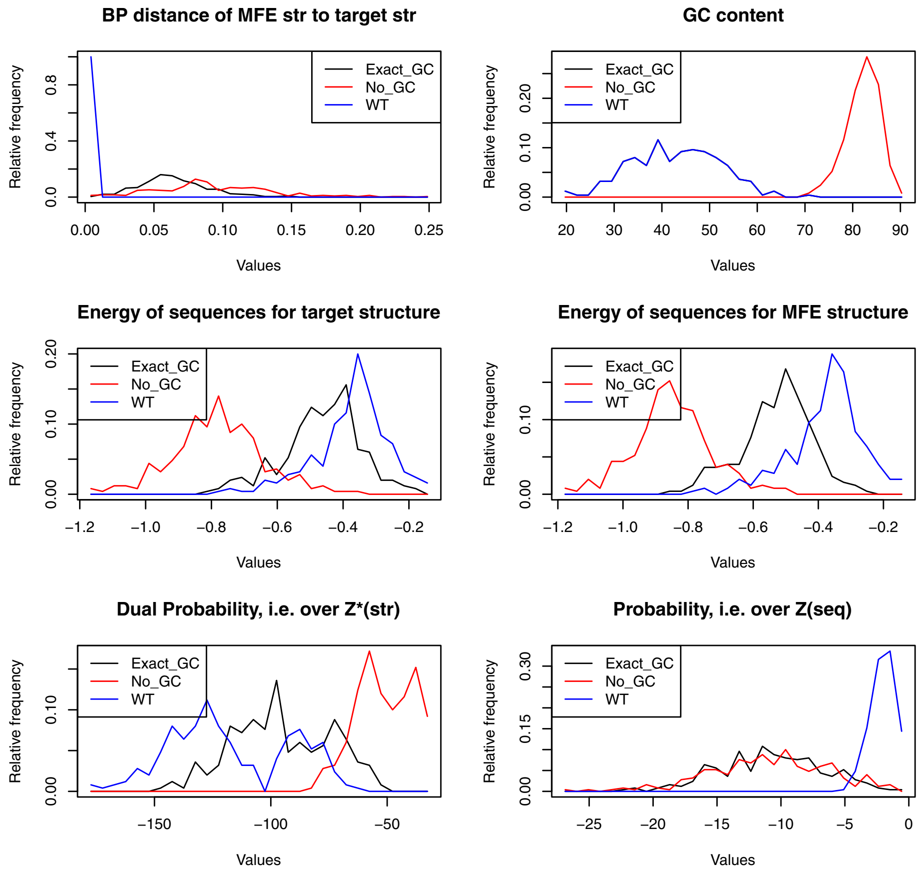

**Fig. 7** Additional measures for the data described in the previous Fig. 6. Density plots are shown for (1) the base pair distance between the minimum free energy (MFE) structure and the target structure, (2) the GC-content, (3) the free energy $E(\mathbf{a}, s_0)$ of the RNA sequences $\mathbf{a}$ with respect to the target structure $s_0$, (4) the free energy $E(\mathbf{a}, s_a)$ of each sequence with respect to its own minimum free energy (MFE) structure, (5) the log dual probability $p^*(s_0) = \frac{\sum_{\mathbf{x}} \exp(-E(\mathbf{x},s_0)/RT)}{Z^*(s_0)}$, and (6) the log probability $p(\mathbf{a}) = \frac{\sum_s \exp(-E(\mathbf{a},s)/RT)}{Z(\mathbf{a})}$

miRBase 20, showing that when GC-content is controlled, sequences sampled by RNAdualPF satisfy the following: (1) the average length-normalized base pair distance between the minimum free energy and target structures is ≈ 0.05, (2) wild type and RNAdualPF sampled sequences have similar free energy with respect to the wild type target structure, (3) as well as similar minimum free energy, (4) similar dual probability, and (5) similar probability to wild type RNA sequences. Taken together, this data shows that if GC-content is controlled, then RNAdualPF returns sequences whose low energy structures tend to resemble the target structure. Figures 8 and 9 are similar to Figs. 6 and 7, except that for each *C. elegans* pre-miRNA, 500 sequences were generated by RNAdualPF, each of whose MFE structure is *identical* to the wild type target structure (this was done by repeatedly sampling sequences from RNAdualPF until 500 sequences were found, that fold exactly into the target pre-miRNA structure). Taken together, Figs. 6, 7, 8 and 9 present convincing evidence that RNAdualPF generates sequences that (approximately) fold into the user-specified target structure, hence supporting our finding that *C. elegans* precursor microRNAs are statistically non-robust, contrary to the finding of [1].

Additionally, we have shown that natural RNAs from the Rfam 12.0 database have *higher* minimum free energy than expected, thus supporting our results in [9] which suggest that functional RNAs are under evolutionary pressure to be only marginally thermodynamically stable. The applications described in this paper demonstrate that RNAdualPF is a useful and extremely fast software tool for evolutionary and synthetic biology.

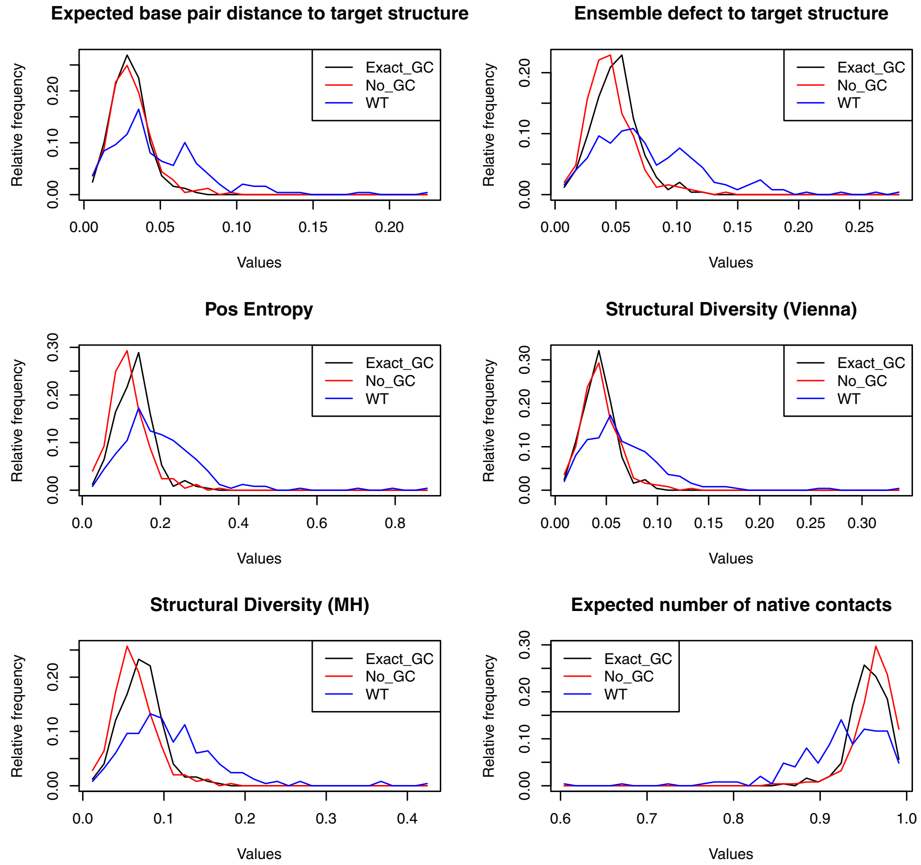

**Fig. 8** For each of the 250 *C. elegans* precursor microRNAs from miRBase 20 and for each of the following cases (**a**), indicated in *black*, and (**b**), indicated in *red*, RNAdualPF generated 500 sequences, whose minimum free energy structure was *identical* to that of the corresponding wild type pre-miRNA (obtained by repeatedly generating samples with RNAdualPF until 500 sequences were found that folded exactly into the target structure). Case (**a**) - *black* lines: All RNAdualPF sequences have GC-content exactly equal to that of the Rfam sequence (Exact GC). Case (**b**) - red lines: RNAdualPF was used in default mode, without controlling GC-content (No GC). Case (**c**) - *blue* lines: wild type (WT) *C. elegans* data. Density plots are shown for (1) the expected base pair distance to target structure $s_0$ [9], (2) the ensemble defect to target structure $s_0$ [20], (3) the positional entropy [22], (4) Vienna structural diversity (called ensemble diversity in [14]), (5) Morgan-Higgs diversity [23], (6) expected proportion of native contacts (called ensemble neutrality in [21]). All measures were normalized by sequence length

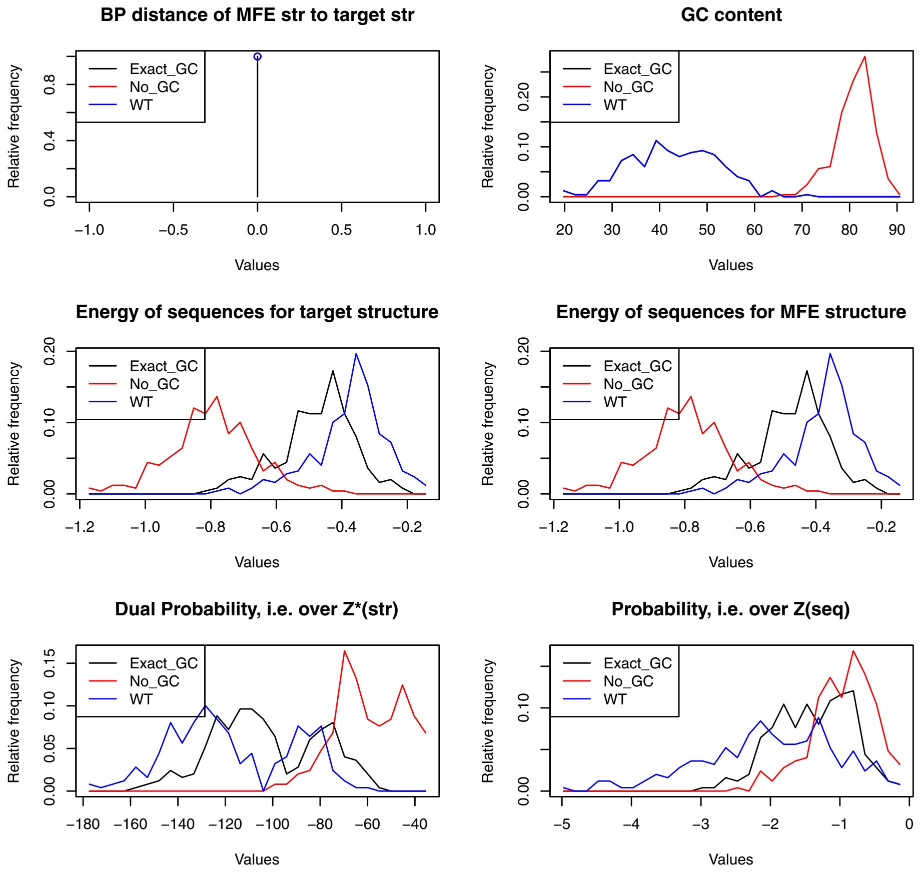

**Fig. 9** Additional measures for the data described in the previous Fig. 8. Density plots are shown for (1) the base pair distance between the minimum free energy (MFE) structure and the target structure, (2) the GC-content, (3) the free energy $E(\mathbf{a}, s_0)$ of the RNA sequences $\mathbf{a}$ with respect to the target structure $s_0$, (4) the free energy $E(\mathbf{a}, s_a)$ of each sequence with respect to its own minimum free energy (MFE) structure, (5) the log dual probability $p^*(s_0) = \frac{\sum_{\mathbf{x}} \exp(-E(\mathbf{x},s_0)/RT)}{Z^*(s_0)}$, and (6) the log probability $p(\mathbf{a}) = \frac{\sum_{s} \exp(-E(\mathbf{a},s)/RT)}{Z(\mathbf{a})}$

## Endnote

[1] When dangling positions are not included in the computation (-d0), the algorithm clearly requires linear time. When dangling positions are included (-d2), run time is exponential in the number of components of the largest multilooop; however, in practice the algorithm is extremely fast, and it is possible to modify the algorithm to always run in linear time.

## Additional file

**Additional file 1:** Supplementary Information. Title is *Supplementary Information for RNAdualPF: software to compute the dual partition function with sample applications in molecular evolution theory*. This is a 3-page PDF file containing a tricky derivation for an efficient computation of the number of external loops of size $N$ with GC-content $k$, where the user can stipulate that certain positions are constrained to contain nucleotides consistent with IUPAC codes. (PDF 273 kb)

### Abbreviations

MFE: Minimum free energy; PLMVd: Peach latent mosaic viroid; sncRNA: Small noncoding RNA

### Acknowledgements

We would like to thank the anonymous reviewers for helpful comments.

### Funding

Funding for this research was provided by National Science Foundation grant DBI-1262439. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

### Availability of data and materials

Source code for OSX and Linux of the C/C++-software `RNAdualPF` is available at http://bioinformatics.bc.edu/clotelab/RNAdualPF under GPL3 licence.

### Authors' contributions

Project design PC. Algorithm design PC, ID, JAGM. Implementation JAGM. Software testing and benchmarking JAGM using a program to test structural diversity implemented by AHB. Manuscript preparation PC, JAGM. All authors read and approved the final manuscript.

### Competing interests

The authors declare that they have no competing interests.

### Consent for publication

Not applicable.

### Ethics approval and consent to participate

Not applicable.

### Author details

[1] Biology Department, Boston College, 140 Commonwealth Avenue, 02467 Chestnut Hill, MA, USA. [2] Research Programme on Biomedical Informatics (GRIB), Department of Experimental and Health Sciences, Universitat Pompeu Fabra, IMIM (Hospital del Mar Medical Research Institute), Dr. Aiguader, 88, Barcelona, Spain. [3] Present Address: Systems Biology Program Centro Nacional de Biotecnología Consejo Superior de Investigaciones Científicas (CSIC) C/ Darwin 3, 28049 Madrid, Spain.

Received: 24 May 2016 Accepted: 26 September 2016
Published online: 19 October 2016

### References

1. Borenstein E, Ruppin E. Direct evolution of genetic robustness in microRNA. Proc Natl Acad Sci. 2006;103(17):6593–598. doi:10.1073/pnas.0510600103.
2. Lorenz R, Bernhart SH, Honer Zu Siederdissen C, Tafer H, Flamm C, Stadler PF, Hofacker IL. ViennaRNA Package 2.0. Algorithms Mol Biol. 2011;6:26. doi:10.1186/1748-7188-6-26.
3. Rodrigo G, Fares MA. Describing the structural robustness landscape of bacterial small RNAs. BMC Evol Biol. 2012;12(1):1–12. doi:10.1186/1471-2148-12-52.
4. Garcia-Martin JA, Dotu I, Clote P. RNAiFold 2.0: a web server and software to design custom and rfam-based RNA molecules. Nucleic Acids Res. 2015;43(W1):513–21. doi:10.1093/nar/gkv460.
5. Los Alamos HIV database. 2015. http://www.hiv.lanl.gov/. Accessed 30 Dec 2015.
6. Krol J, Sobczak K, Wilczynska U, Drath M, Jasinska A, Kaczynska D, Krzyzosiak WJ. Structural features of microRNA (miRNA) precursors and their relevance to mirna biogenesis and small interfering RNA/short hairpin RNA design. J Biol Chem. 2004;279(40):42230–2239. doi:10.1074/jbc.M404931200.
7. Hofacker IL, Fontana W, Stadler PF, Bonhoeffer LS, Tacker M, Schuster P. Fast folding and comparison of RNA secondary structures. Monatsch Chem. 1994;125:167–88.
8. Zadeh JN, Wolfe BR, Pierce NA. Nucleic acid sequence design via efficient ensemble defect optimization. J Comput Chem. 2011;32(3):439–52.
9. Garcia-Martin JA, Clote P, Dotu I. RNAiFold: a constraint programming algorithm for RNA inverse folding and molecular design. J Bioinform Comput Biol. 2013;11(2):1350001. doi:10.1142/S0219720013500017.
10. Ding Y, Chan CY, Lawrence CE. Sfold web server for statistical folding and rational design of nucleic acids. Nucleic Acids Res. 2004;32:0.
11. Kozomara A, Griffiths-Jones S. mirbase: annotating high confidence microRNAs using deep sequencing data. Nucleic Acids Res. 2014;42(Database):68–73. doi:24275495.
12. Reinharz V, Ponty Y, Waldispuhl J. A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. Bioinformatics. 2013;29(13):308–15.
13. Turner DH, Mathews DH. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. Nucleic Acids Res. 2010;38(Database):280–2. doi:10.1093/nar/gkp892.
14. Gruber AR, Bernhart SH, Hofacker IL, Washietl S. Strategies for measuring evolutionary conservation of RNA secondary structures. BMC Bioinforma. 2008;9(1):1–19. doi:10.1186/1471-2105-9-122.
15. McCaskill JS. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Biopolymers. 1990;29: 1105–1119. doi:10.1002/bip.360290621.
16. Ding Y, Lawrence CE. A statistical sampling algorithm for RNA secondary structure prediction. Nucleic Acids Res. 2003;31:7280–301.
17. Busch A, Backofen R. INFO-RNA, a fast approach to inverse RNA folding. Bioinformatics. 2006;22(15):1823–31. doi:10.1093/bioinformatics/btl194.
18. Zuker M, Mathews DH, Turner DH. Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide In: Barciszewski J, Clark BFC, editors. RNA Biochemistry and Biotechnology. NATO ASI Series. Dordrecht: Kluwer Academic Publishers; 1999. p. 11–43.
19. Griffiths-Jones S, Bateman A, Marshall M, Khanna A, Eddy SR. Rfam: an RNA family database. Nucleic Acids Res. 2003;31(1):439–41.
20. Dirks RM, Lin M, Winfree E, Pierce NA. Paradigms for computational nucleic acid design. Nucleic Acids Res. 2004;32(4):1392–1403. doi:10.1093/nar/gkh291.
21. Pei S, Anthony JS, Meyer MM. Sampled ensemble neutrality as a feature to classify potential structured RNAs. BMC Genomics. 2015;16(1):1–12. doi:10.1186/s12864-014-1203-8.
22. Huynen M, Gutell R, Konings D. Assessing the reliability of RNA folding using statistical mechanics. J Mol Biol. 1997;267(5):1104–12. doi:10.1006/jmbi.1997.0889.
23. Morgan SR, Higgs PG. Barrier heights between ground states in a model of RNA secondary structure. J Phys A: Math Gen. 1998;31:3153–170. doi:10.1088/0305-4470/31/14/005.
24. Nawrocki EP, Burge SW, Bateman A, Daub J, Eberhardt RY, Eddy SR, Floden EW, Gardner PP, Jones TA, Tate J, Finn RD. Rfam 12.0: updates to the RNA families database. Nucleic Acids Res. 2015;43(D1):130–7. doi:10.1093/nar/gku1063.
25. Sprinzl M, Horn C, Brown M, Ioudovitch A, Steinberg S. Compilation of tRNA sequences and sequences of tRNA genes. Nucleic Acids Res. 1998;26(1):148–53. doi:10.1093/nar/26.1.148.
26. Juhling F, Morl M, Hartmann RK, Sprinzl M, Stadler PF, Putz J. tRNAdb 2009: compilation of tRNA sequences and tRNA genes. Nucleic Acids Res. 2009;37(Database):159–62.
27. Garcia-Martin JA, Clote P. RNA Thermodynamic Structural Entropy. PLoS ONE. 2015;10(11):0137859. doi:10.1371/journal.pone.0137859.
28. Garcia-Martin JA. RNA inverse folding and synthetic design. Ph.D. dissertation in Biology, Boston College. 2016. Dissertation made available on June 28, 2016 and will remain accessible indefinitely: http://hdl.handle.net/2345/bc-ir:106989.