Journal of Heuristics
https://doi.org/10.1007/s10732-021-09471-3

# Qfold: a new modeling paradigm for the RNA folding problem

**Mark W. Lewis**[1] · **Amit Verma**[1] · **Todd T. Eckdahl**[2]

Received: 10 July 2020 / Revised: 25 November 2020 / Accepted: 10 February 2021


**Abstract**
Ribonucleic acid (RNA) molecules play informational, structural, and metabolic roles in all living cells. RNAs are chains of nucleotides containing bases {A, C, G, U} that interact via base pairings to determine higher order structure and functionality. The RNA folding problem is to predict one or more secondary RNA structures from a given primary sequence of bases. From a mathematical modeling perspective, solutions to the RNA folding problem come from minimizing the thermodynamic free energy of a structure by selecting which bases will be paired, subject to a set of constraints. Here we report on a Quadratic Unconstrained Binary Optimization (QUBO) modeling paradigm that fits naturally with the parameters and constraints required for RNA folding prediction. Three QUBO models are presented along with a hybrid metaheuristic algorithm. Extensive testing results show a strong positive correlation with benchmark results.

**Keywords** RNA folding · Structure prediction · QUBO · Path relinking

## 1 Introduction

QUBO is a general purpose modeling framework that has been applied to many specific areas and is currently the primary structure required by quantum annealing computers (Boothby and Roy 2016; Choi 2008). In this report, we present QUBO models for solving the RNA folding problem, which seeks to predict how base pairing determines RNA secondary structures that enable biological functions associated with information flow, structure, and metabolism.

✉ Mark W. Lewis
mlewis14@missouriwestern.edu

[1] Craig School of Business, Missouri Western State University, 4525 Downs Dr., Saint Joseph, MO 64507, USA

[2] Department of Biology, Missouri Western State University, 4525 Downs Dr., Saint Joseph, MO 64507, USA

Published online: 06 March 2021

We leverage the conventional approach of optimizing the minimum free energy (MFE) associated with an RNA secondary structure with the understanding that a given solution to the problem does not necessarily imply a single secondary structure. Gardner and Giegerich (2004) suggest that asking for a single structure to be a consensus for a family of structures is the wrong question and Mathews (2019) concludes that base pairs fluctuate, resulting in secondary structure models that represent alternative states in equilibrium with each other. Thus while MFE is a good, albeit imperfect, objective function metric to optimize, sub-optimal solutions can also be useful predictors of secondary structure.

Knowing that an exact solution is not necessarily better than a collection of sub-optimal ones, and considering the combinatoric nature of the problem, the obvious direction is a heuristic approach. The heuristic presented here generates a set of variables representing feasible RNA stem (also known as helix) fragments, from which a QUBO instance is generated and then solved with a new, generic hybrid QUBO solver. The solver combines a greedy 1-flip search with path relinking, backtracking and strategic oscillation around local optima. The software and dataset is available for download on GitHub (Verma 2020).

The QUBO solution process is not customized for the specific problem of RNA folding, however crafting specific problem characteristics into the QUBO model requires some subject matter expertise. Even though any mixed integer programming (MIP) solver, such as Cplex or Gurobi, can be used to solve a QUBO instance, binary *quadratic* problems, especially those with a dense interaction matrix, generate extremely large mixed integer *linear* programs. In addition, the linear programming relaxations used to reduce the search tree are generally weak (have a large MIP gap), so that the suboptimal solutions found are generally not of high quality. Another QUBO solution approach is to use hardware specifically designed to solve QUBO problems, such as D-Wave Systems Quantum Annealer (D-Wave Systems 2020) or Fujitsu's Digital Annealer (Fujitsu 2020).

In this paper we report on a new approach for a difficult and well-studied problem in computational biology that has not previously been modeled and solved as QUBO. Section 2 presents a survey of the RNA folding literature and QUBO. Section 3 describes three QUBO models and algorithm pseudocode followed by computational testing results in Sect. 4 and then Sect. 5 contains conclusions.

## 2 Literature review

Predicting the secondary structure of an RNA sequence is known to be NP-Hard (Saad et al. 2012). RNA folding is often solved by dynamic programming (DP) based methods (Zuker and Stiegler 1981) with a major limitation being that the runtime scales cubically with RNA length and many variants have been proposed to address this issue. For instance, Huang (2019) used beam search in conjunction with DP to report improved accuracy for long RNA molecules The free energies associated with base pairs are used to predict RNA structure, and are often incorporated into a model as static numbers even though they are not known with certainty. To

address this, Yan et al. (2020) have recently developed a graph neural network using an adjacency matrix of base-pairing probabilities.

Many scholars have attempted to integrate big data modeling paradigms with RNA folding. For example, Zhang et al. (2019) combined a convolution neural network with DP. They developed a technique to classify large scale data to predict the base pairing probability and achieved a 30% higher success rate. Singh et al. (2019) used a two-dimensional deep neural network and transfer learning for base pair prediction, including non-canonical base pairs and pseudoknots. The model trained with more than 10,000 non-redundant RNAs and achieved a statistically significant improvement in prediction accuracy compared to traditional methods. For a recent survey on RNA folding algorithms, we refer the reader to Fallmann (2017).

The RNA folding problem has also received attention from the quantum computing researchers. Shi (2019) propose a quantum assisted genetic algorithm to predict RNA folding. Their algorithm involved multiple populations evolving through genetic exchange performed by a transfer operator. The authors concluded that the technique improves the prediction accuracy and sensitivity for medium length RNA sequences.

QUBO research has its origins in the 1960s, when it focused on pseudo-Boolean optimization (Hammer and Rudeanu 1968) and constrained models. When a quadratic binary problem involves constraints, it can be recast into an equivalent unconstrained model using quadratic infeasibility penalty terms (Kochenberger et al. 2004). In some cases, additional variables are needed to enforce the constraint while in others, such as the models we report here, no additional variables are needed. In 1988, the important physics problem known as Ising Spin Glass was formulated as a QUBO using $+1/-1$ variables to denote the spin states (Barahona et al. 1988). As a modeling note, $+1/-1$ variables are easily converted to 0/1 binary variables. Four problems in computational molecular biology (Multiple Sequence Alignment, Lattice Protein Folding, Contact Map Overlap, and Rotamer Assignment) are presented as QUBO models in Forrester and Greenberg (2008) along with linearization techniques to remove the quadratic components of the model in order to facilitate solutions with a general purpose solver. A good survey of quadratic binary optimization is provided in Kochenberger et al. (2014) while Lucas (2014) shows the general applicability of QUBO modeling by providing details on QUBO formulation of many of the well-known NP-Hard problems and a tutorial on QUBO models and their modern applications appears in Glover et al. (2019).

Small QUBO models can be solved exactly (Pardalos and Rodgers 1990; ILOG 2019) but because they are NP-Hard (Pardalos and Jha 1992), heuristic approaches are needed for large and dense instances (typically over ~1000 variables and over 20% dense). The QUBO problems solved in this paper contain up to 25,000 variables at about 40% density. Large problems can require a significant amount of memory and processing power so that methods to reduce the size include partitioning heuristics via graph clusters (Mauri and Lorena 2012). One-pass heuristics can also be used to speed up processing of large QUBO (Glover et al. 2002). Preprocessing of large QUBO can reduce their size by discovering variables that can be fixed to 0/1 or that have a fixed relationship to another variable such that these variables are eliminated from the original model (Glover et al. 2018). The best known solution

technique for QUBO combines tabu search with path relinking (Wang et al. 2012) which has been demonstrated to quickly find the best known solutions to widely used benchmark QUBO datasets (Beasley 1990; Palubeckis 2006). The QUBO modeling approach is the foundation for the D-Wave Systems quantum annealing computer (D-Wave Systems 2020) and Fujitsu's commercial digital annealer (Fujitsu 2020).

## 3 QUBO models for secondary structure prediction

The fundamental building blocks of both RNA and DNA are nucleotides (nt), often referred to as bases, which are composed of a sugar, a phosphate, and a nitrogenous base. The only chemical difference between DNA and RNA is the type of sugar used, which is deoxyribose for DNA and ribose for RNA. Nucleotides are strung together to form DNA or RNA polynucleotides by chemical reactions that result in a backbone of alternating sugars and phosphates. The most common nitrogenous bases found naturally in DNA are adenine (A), cytosine (C), guanine (G), and thymine (T). In naturally occurring RNA, T is replaced by uracil (U). Bases in DNA and RNA occur in ordered sequences that have the informational content required for the expression of genes, and in the case of RNA, for additional structural and metabolic functions. The ordered sequence of nucleotides is known as the *primary* structure, defined as an ordered sequence ($S = b_1, b_i, \ldots, b_n$) of length $n$ with bases $b \in \{A, C, G, U\}$. The sequence S is the primary input needed for *secondary* structure prediction.

The most important behavior of nucleic acids in terms of understanding their functions in nature or using them for biotechnology is base pairing. Base pairing refers to the formation of weak hydrogen bonds between two bases. Bases pair with each other according to canonical pairings first described by Watson and Crick (1953). Watson–Crick base pairing rules are that A pairs with T or U and C pairs with G. Additionally, U and G can pair as non-canonical or "wobble base pairs." When successive bases engage in base pairing, single strands of DNA or RNA become double stranded. In RNA a single strand folds into *secondary* structures such as stems (helices), hairpins, bulges, and junctions. Figure 1 illustrates a primary RNA sequence of bases and a secondary structure predicted by the RNA folding software ViennaRNA (Kerpediev et al. 2015) that includes a stem of 6 base pairs and a hairpin loop of 6 unpaired bases. The *tertiary* structure of RNA is its three-dimensional shape and is not addressed in this paper.

The strength of each base pair interaction can be measured experimentally and we have used those reported by Turner and Mathews (2009). We have adopted the common practice in RNA modeling of using the Minimum Free Energy (MFE) as a measure of the stability of RNA secondary structure. In other words it is assumed an initial strand of RNA will fold and pair with itself in a way that avoids free energy in the secondary structure's end state. In addition to Watson–Crick canonical base pairing, G pairing with U is allowed. Three other rules help determine a set of base pairs that can be used as binary variables in an optimization. First, a base can only pair with one other base (not multiple bases). Second, the sugar and phosphate backbone of RNA is generally not flexible enough to allow pairing between bases having less than $d$

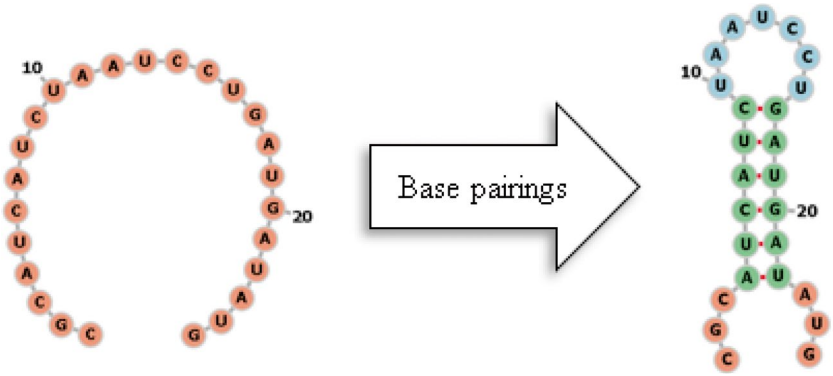


**Fig. 1** An example of a RNA primary structure (left) and the secondary structure predicted by RNA folding software (Vienna RNA Web Service 2020) (right). Color code: green = stem (helix); orange = unpaired bases; yellow = interior loops; blue = hairpin loops (Color figure online)

intervening bases. A value of $d=4$ is commonly used because hairpin loops with 3 of fewer unpaired sequential bases cannot flex enough to make a complete turn. Finally, base pairings should not cross, e.g. a base pair in Fig. 1 between A4 (base A in sequence position 4) and U22, denoted as A4:U22 is a feasible nested pair with A5:U21. While A4:U21 and A5:U22 are feasible base pairs *individually*, they are not feasible *nested* pairs. Nested pairs create stable structures (sometimes called stacked quartets) that tend to minimize free energy. Two base pairs $\left(i^1, j^1\right)$ and $\left(i^2, j^2\right)$ are sequential nested pairs if their sequential positions satisfy $i^1 + 1 = i^2$ and $j^2 + 1 = j^1$. Figure 2 illustrates the difference between non-crossing and crossing base pairing. There are no base pairings that cross in the left illustration, while base pairs (4, 18) and (8, 22) are crossing on the right. These three rules, along with the base pair weights provided by Turner and Mathews (2009) appearing in the objective function, are the basic elements of our optimization models.

QUBO models have the general form

$$\text{Max:} \quad \sum_{i}^{n} c_i x_i + \sum_{i}^{n} \sum_{j}^{n} c_{ij} x_i x_j, \text{or equivalently Max } x' Q x$$

where $x$ is binary and Q is a symmetric $n \times n$ matrix of coefficients $c_{ij}$. The definition and meaning of the binary variables $x_i$ and the coefficients $c_i$ and $c_{ij}$ are critical to accurate modeling. The diagonal of Q contain linear terms $c_{ii}$ that quantify the effect of flipping a single variable from 0 to 1 without regard to possible interactions with other variables and the off-diagonal quadratic coefficients $c_{ij}$ represent

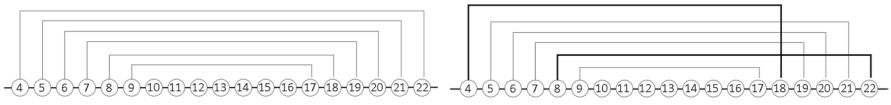

**Fig. 2** Non-crossing (left) and crossing base pairs (bold lines)

the magnitude of the interaction effect when both variables $x_i$ and $x_j$ are set to one. Without loss of generality, $x_i x_j = x_j x_i$ resulting in a symmetric Q matrix where we need only define pairs $x_i x_j$ for $i < j$.

We will consider three models with differing variable definitions and abilities. Model 1 is provided as an example of a very simple model that may appear reasonable but has deficiencies. In Model 1 let $x_i \in \{0,1\}$ indicate whether the base $b$ at position $i$ in the sequence of length $n = |S|$ is paired with any other base. The strength of the interaction between bases in positions $i$ and $j$ in S is denoted by the coefficient $c_{ij}$. Thus if it is important to have both $i$ and $j$ be part of a base pair, then $c_{ij}$ is set to a positive number indicating a bonus in the objective function and a similar statement is made if they should not be in a base pair. However, this variable definition and model will not yield a solution indicating which bases are being paired and does not allow crossing pair constraints, hence a more accurate model is needed.

A more accurate $x'Qx$ model will identify which base is paired to another and also include constraints such as the singularity constraint $\sum_k^n x_k \leq 1 \forall i \in S$, which restricts a variable to one base pair, and crossing pair constraints, defined as two matched pairs $(i, j)$ and $\left(i', j'\right)$, where $\left(i < i' < j < j'\right)$. In order to avoid selecting base pairs that cross, additional constraints are needed and these will also be incorporated into the objective function to yield unconstrained $x'Qx$.

In general, any set of constraints $Ax \leq b$ may be transformed to the objective function to yield an equivalent unconstrained form Kochenberger et al. (2004). The conversion requires additional slack variables, one for each inequality constraint, to convert to the equality sense needed to generate associated penalty terms in the objective function:

$$\text{Max:} \quad \sum_i^n \sum_j^n c_{ij} x_i x_j + P(Ax - b)^t (Ax - b)$$

where P is a scalar penalty term, $A$ is the matrix of constraint coefficients and $b = 1$.

In Model 2, a variable $x_k^{i,j}$ no longer represents a single base, but a feasible base pair $(i, j)$, where $i$ and $j \in S$. Thus, $x_k^{i,j} \in \{0, 1\}$ indicates whether the base $b$ at position $i$ in the sequence of length $n = |S|$ is paired with base $b$ at position $j$ and an enumeration of feasible base pairs meeting minimum distance and base pairing requirements is required in order to build a problem instance. Let $P = ((b_1, b_2), \ldots, (b_i, b_n))$ be an ordered sequence of feasible pairings from set S with worst case $|P| = n * (n-1)$ and the index $k$ in $x_k^{i,j}$ varies from one to |P|. The linear term $c_k$ is the magnitude of the weight associated with the base pair $k$ and the interaction between nested pairs $k$ and $l$ is $c_{kl}$. The quadratic coefficients $c_{kl}$ are the negative of the thermodynamic free energies published in Turner and Mathews (2009) in order to match the maximization sense of the QUBO objective function.

Penalty terms in the objective are used to implement both the singularity and non-crossing constraints. If a feasible base pair $k$ crosses with another base pair $l$, then the quadratic penalty term $c_{kl} = M^-$, where $M^-$ is a large negative number that serves to prevent the selection of both $x_k$ and $x_l$ (where the superscripts are removed for brevity). Likewise, a quadratic penalty term also prevents the selection of the

same base in more than one base pair. When $x_k x_l = 1$ a feasible nested pair (stacked quartet) becomes part of the predicted fold. Although this model is simple, it incorporates all the necessary constraints. As a comparison, using mixed integer *linear* programming (no quadratic variables) as the modeling paradigm requires two constraints for each pair of variables (Gusfield 2019) which generates a large number of constraints, e.g. a Q with $n = 1500$ and 50% dense would generate over a million constraints.

This second model incorporates all three sets of constraints (base pairing, singularity, emphasis of nested pairs). However, it generates a large number of variables because all feasible parings of two bases are enumerated. Another disadvantage of this model is that by defining a variable as a *single* base pair allows *unpaired bases on either side* of the pair creating a helix (stem) of length one, sometimes called "lonely pairs". This is not a good modeling practice for creating stable secondary structures since single base pairs tend to destabilize the secondary structure (Findeiss et al. 2018).

Model 3 addresses both size and stem length concerns. It dramatically reduces the number of variables generated by having the variable $x_k$ represent a sequential *nested* pair (stacked quartet) of 4 bases, hence $x_k^{i^1,j^1,i^2,j^2} \{0,1\}$ indicate the pairs $(i^1, j^1)$ and $(i^2, j^2)$ are both feasible and nested, i.e. $i^1 + 1 = i^2$ and $j^2 + 1 = j^1$. The four superscripts are removed for brevity in future references. The linear terms used for all combinations of feasible stacking are shown in Fig. 3 and are taken from (Mathews 2020; Turner and Mathews 2009). The rows of Fig. 3 are the first base pair of $x_k$ and the columns the second. The quadratic interaction terms between stacked quartets are a bonus $M^+$ used to promote nesting and sequencing with other nested pairs in order to generate long stable stems. $M^-$ prevents crossing nested pairs and assigning a base to more than one nested pair.

As an example of the use of $M^+$ and $M^-$, consider three variables from Fig. 1, where the base type has been added to the number in the sequence. Let $x_1 = \{(4A, 22U), (5U, 21A)\}$, $x_2 = \{(5U, 21A), (6C, 20G)\}$ and $x_3 = \{(4A, 19U), (5U, 18A)\}$ represent a subset of the possible nested pairs. In the Q matrix, $c_{1,2} = M^+$ to support a long stem formed by $x_1$ and $x_1$ while $c_{1,3} = M^-$ to penalize assigning 4A to more than one nested pair.

| | | Stacked pair 5' to 3' | | | | | |
|---|---|---|---|---|---|---|---|
| | | AU | CG | GC | UA | GU | UG |
| | AU | 0.9 | 2.2 | 2.1 | 1.1 | 0.6 | 1.4 |
| | CG | 2.1 | 3.3 | 2.4 | 2.1 | 1.4 | 2.1 |
| 5' | GC | 2.4 | 3.4 | 3.3 | 2.2 | 1.5 | 2.5 |
| to | UA | 1.3 | 2.4 | 2.1 | 0.9 | 1 | 1.3 |
| 3' | GU | 1.3 | 2.5 | 2.1 | 1.4 | 0.5 | -1.3 |
| | UG | 1 | 1.5 | 1.4 | 0.6 | -0.3 | 0.5 |

**Fig. 3** Stacked Quartet free energies (kcal/mol) from Turner and Mathews (2009)

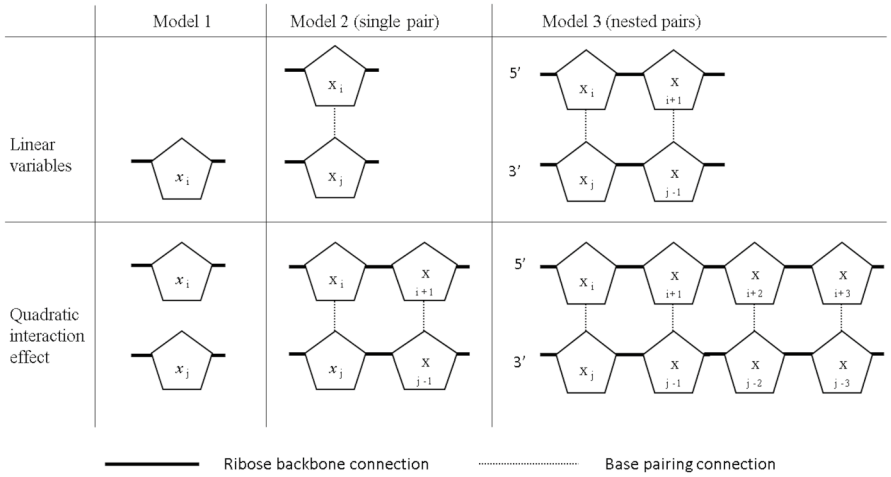


**Fig. 4** Illustration of the binary variables and their quadratic interaction

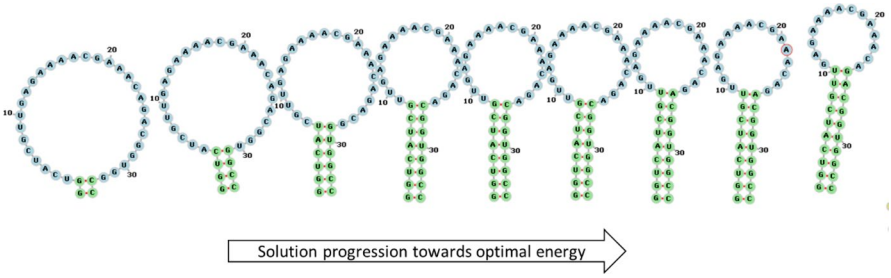


**Fig. 5** Qfold solution progression to optimal free energy via stem growth

Figure 4 shows the variables used in the three models and the meaning of their quadratic interactions. Model 3 reduces the number of variables by approximately one-third compared to Model 2, incorporates all constraints via penalties, eliminates the possibility of single pairs, and supports the creation of long stems. Because the objective function includes $M^+$ terms that are not derived from experimental observations of base pair energies, we differentiate the QUBO objective value measure from MFE by denoting it QMFE. A comparison of QMFE and MFE results presented in Sect. 4 showed a strong positive correlation between the two measures.

As an example of the output provided by Qfold consider Fig. 5 showing a progression of solutions from left to right representing a progression from high to low free energy for the molecule PDB_00727 (length = 34 bases). The problem actually starts with no bases paired, then grows the central stem feature by selecting variables that optimize QMFE.

A pseudoknot is a complex stem-loop structure in which half of one stem is between the two halves of a second stem because of crossing base pairs (see Fig. 2). Pseudoknots are often disallowed in RNA secondary structure prediction because

they complicate the prediction process by increasing the number of possible interactions to consider and because they cross the boundary from two-dimensional to three-dimensional RNA modeling. However, pseudoknots involving the feasible base pairs generated can be incorporated by simply removing the penalty $M^-$ or lowering its magnitude to represent a soft constraint wherein base pair crossing is an allowable option if it improves the objective function by an amount greater than the soft penalty incurred.

### 3.1 Pseudocode

Qfold is the term we use to denote the combination of variable enumeration and Q matrix generation followed by solution via QUBOsearch. At a high level, the process enumerates from a sequence S the feasible base pairs greater than *min_d* minimum distance and generates a Q matrix.

```
Qfold ( S, min_d, base_pairing_weights, time_limit, back_track, repeat, DOE )
       Q ← Q_generation ( S, min_d, base_pairing_weights )
       Solutions ← QUBO_search ( Q, time_limit, back_track, repeat, DOE )
```

#### 3.1.1 Enumeration of feasible nested pairs

The two main functions of Qfold are to generate the Q matrix from the input RNA sequence and to calculate the solution to Max: $x'Qx$. The coefficients comprising the Q matrix are based on the variables generated and their interaction effects. The variables generated are determined by nesting of feasible base pairings.

A Q matrix is generated by first discovering feasible base pairs meeting the minimum distance (min_d) requirement. This $O(n^2)$ operation is performed only once and consists of checking if base $i$ will pair with base $j = i + \text{min\_d}$ up to the length $n$ of S. The calculation of the variables representing nested pairs is similar. The interaction weights between nested pairs (Fig. 3) and the $M^+$ and $M^-$ terms are computed with the output being a Q matrix in (row, col, value) format.

```
Q_generation ( S, min_d, weights )
       P ← Determine_feasible_pairs ( S, min_d )
       Q ← Generate_nested_pairs ( P, weights )
```

The set P contains all feasible pairings $(i, j)$ where, for example a feasible pair is $S_i$ is a C nucleotide in position $i$ and $S_j$ is a G in position $j$. A feasible pair satisfies the minimum distance criteria allowed between pairs and is either A-U, U-A, C-G, G-C, U-G or G-U. In other words, A does not pair with C or with G. RNA is described as having a starting nucleotide designated as 5′ and an ending nucleotide designated 3′. In our looping structures below, the index variables start at the 5′ beginning of the strand sequence and moves towards the ending 3’.

```
Determine_feasible_pairs ( S, min_d )
    k = 1
    for i = 1 to |S| - min_d
        for j = i + min_d to |S|
            if Si can pair with Sj then          // store gene location in P
                P[k][1] = i
                P[k][2] = j
    return P
```

The matrix Q contains information on all sequential nested pairings of feasible base pairs P. The interaction bonus between nested base pairs is $M^+$ and penalties $M^-$ ensure that a gene at position *i* is only paired with one other base (or none) and that crossing pairs are not selected.

```
Generate_nested_pairs ( P, weights )
    k = 1
    for i = 1 to |P|
        for j = i + 1 to |P|
            if Pi can nest with Pj then
                Q[i][i] = Turner_weight_lookup ( P, i, j )
                Q[i][j] = M+
            if Pi shares a gene location with Pj then
                Q[i][j] = M-
            if Pi creates a crossing pair with Pj then
                Q[i][j] = M-
    return Q
```

### 3.1.2 Solving QUBO

There are many software and hardware products available for solving $x'Qx$. Commercial solvers such as Cplex and Gurobi are powerful mixed integer *linear* programming solvers capable of solving quadratic binary optimizations. D-Wave Systems (2020) and Fujitsu (2020) have developed hardware that specifically solves QUBO problems. A web based QUBO solver based on (Glover et al. 2019) is available at (Meta-Analytics 2020).

As the size and density of the Q matrix increases, the problem tends to become more difficult to solve because it involves NP complexity with exponential growth so that metaheuristics are employed to generate good answers in a reasonable amount of time. The metaheuristic presented in Qfold is a hybrid combining greedy 1-flip neighborhood search, followed by path relinking between elite solutions, backtracking and strategic oscillation to break out of local optima, and restarts after these methods stop yielding improvements. The program also includes some customization involving RNA folding in that it accepts benchmark structures for comparison measures and outputs an RNA sequence with dot-parenthesis data used to visualize

the structure. Qfold also incorporates a one-pass post-processing routine that looks for obvious missed pairs. Qfold is available on GitHub at Verma (2020).

The primary power of QUBO_search is its ability to quickly evaluate the effect of flipping a single bit, $x_i = 1 - x_i$, allowing selection of the variable having the greatest effect on a local solution in O(n) time (Kochenberger et al. 2004). This efficient 1-flip evaluation is a part of all the hybrid components and the main component of the greedy 1-flip neighborhood search, which returns the best (most improving) move. The search reads in a starting solution and then improves upon it until a local optimum is encountered, wherein additional routines try to break out of the locality and improve the solution, and if those fail then a new starting solution is read.

```
QUBO_search ( Q, time_limit, back_track, repeat, DOE )

while ( time_limit not exceeded OR i < max_iterations )
  x^i ← Read_experimental_design ( DOE^i )
  while ( NOT Is_a_Local_optima ( x^i ) )
    x^i ← Greedy_1_flip ( x^i)          // select best improving bit-flip

    if QMFE(x^i) > QMFE*                // update best solution
       QMFE* = QMFE(x^i)
       incumbent_x ← x^i;
       threshold  = 0.9 * QMFE^i

    if QMFE(x^i) > QMFE^i
       QMFE^i = QMFE(x^i)
       best_ x^i ← x^i

    if Is_a_Local_optima ( x^i )        // explore around local optimum
       Path_relinking ( x^i, incumbent_x )
       Back_track ( x^i, back_track, repeat )
       if QMFE(x^i) > threshold  Strategic_Oscillation ( x^i )

  i++                             // restart search with new starting solution
```

A set of good starting solutions can greatly improve the performance of an algorithm tasked with exploring a large solution landscape, especially if the starting solution happens to be close to an optimal. Uniformly random starting solutions are often employed as a default method of providing an approximate cover of the solution landscape, i.e. generating a diverse set of starting solutions. However, as problem size increases, the number of restarts becomes relatively smaller compared to the number of solutions explored during the search. For example, RNA molecule ASE_00001 generated a Q matrix with $n = 4269$ variables and in 60 s over 530,000 solutions were explored using only 113 different starting points, or 0.02% of all solutions evaluated. Given that the number of starting solutions will be relatively small, generating a good set of starting solutions is important.

A set of starting solutions using Design of Experiments (DOE) methods alleviates the problem of a small set of starting points providing a good cover of the landscape. The experimental design approach used is a 2-level *m x n* fractional factorial table where *m* is the number of solutions generated based on the number of variables *n* in Q. For example if $n = 7$ then 8 solutions are initially created and then 8 more

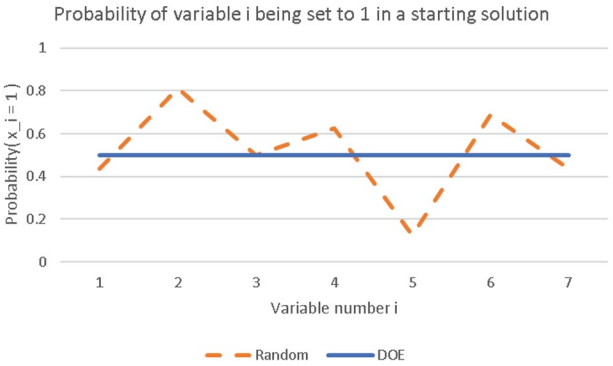


**Fig. 6** Randomly setting variables in starting solution is not uniform when using a small sample size

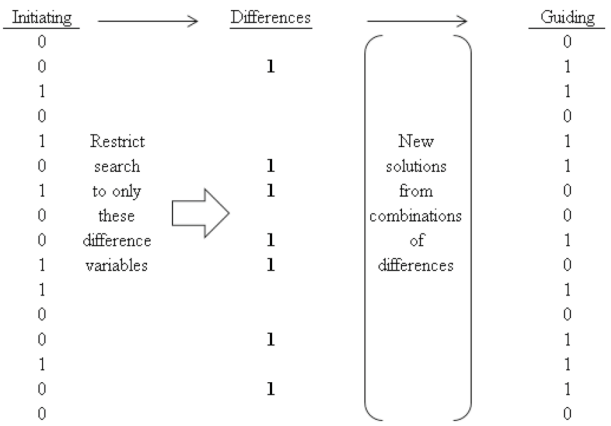


**Fig. 7** Example of Path Relinking Implementation

are generated by taking the complement $(1 - x_i)$ of the first set, producing 16 starting solutions. As illustrated by Fig. 6, the probability of a variable $x_i$ being set to 1 is exactly 50% instead of using a uniform random distribution where, for example, $\text{prob}(x_5 = 1) < 20\%$. Regardless of size, the set of precomputed DOE solutions will always have this equal probability distribution. The experimentally designed solution table is only used for starting solutions and readers are referred to Lewis and Kochenberger (2016) for further details.

Path relinking is a powerful tool for improving solutions via search intensification around elite solutions. The general concept is to start from an initial solution and move towards a guiding one while exploring the reduced solution space consisting of the possible combinations of difference bits. Figure 7 illustrates the concept. In Qfold, path relinking is implemented as a greedy 1-flip search over the difference bits. Path relinking is employed with a long term memory tracking the number of times a bit has been flipped since the program started so that variables that are

repeatedly being flipped are not selectable until their flip count is less than a set percentage of the total count of bit flips.

Backtracking is an approach to break out of a local optima by undoing $k = \text{back\_track}$ number of recent 1-flips for variables $i$, creating a set of variables $B = \{x_i^k\}$ created where $|B| = k$. The concept of backtracking is related to the idea suggested by Laguna and Glover (1993) and Glover (2020) that improving moves are more likely to select attributes of optimal solutions than non-improving ones. Hence backtracking a number of moves from a local optima and restarting the search from that point seems a reasonable way to explore the area near the local optima via ascending moves. Backtracking restarts the search process with a tabu tenure based on the back_track parameter, restricting $x_i^k \in B$ from being selected from the candidate list, thus avoiding repetitions of the same sequence of flips that led to the local optimum encountered. The repeat parameter determines how many times to backtrack before moving on to Strategic Oscillation. A value of back_track close to size $n$ has the effect of restarting the search at the beginning solution, but with tabu tenures based on a short term memory set at the beginning of the backtrack routine. If back_track is small, then the search is confined to the neighborhood space close to the local optimum.

The Strategic Oscillation (SO) routine is entered if the objective of the current solution is within 90% (the critical level) of the current best for the given starting solution, which is not necessarily the current incumbent. The neighborhood around this critical level is searched via a greedy 1-flip where non-improving (destructive) flips are allowed if they are the best possible move. The key difference being that greedy 1-flip in the other areas of QUBO_search will not allow selecting a variable to be flipped if it degrades the current solution. Unlike traditional SO techniques that would leverage problem specific information or constraint limits to gauge when to reverse direction, e.g. filling a knapsack beyond its capacity and then reversing by removing items, our generic SO simply allows destructive moves. SO is terminated after back_track number of iterations.

## 4 Computational testing and results

The RNA secondary STRucture and statistical Analysis Database (RNA STRAND) is a collection of secondary structures determined experimentally or through comparative analysis for a large variety of natural and synthetic RNA molecules (Andronescu et al. 2008). Qfold was tested using the base sequences of over 453 RNAs from RNA STRAND Database (2008). Testing was performed using 64 bit Windows 7 on an 8-core i7 3.4 GHz processor with 16 GB RAM. The Qfold software was developed and compiled in C with Visual Studio 2019 and the testing was coordinated via Python. From a computation standpoint, the best practices for benchmarking RNA secondary structure prediction suggested by Mathews (2019) were used as a guide. The magnitude of the penalty term $M^-$ was set to a relatively large value of $-2000$ to guarantee no sub-optimal solutions are generated that violate constraints. The magnitude of the bonus term $M^+$ used to reward the nesting of nested pairs to create long stems was set to 1.6 which is the average Free Energy

**Table 1** Summary problem characteristics

| | |
|---|---|
| Total number tested | 433 |
| Number experimentally validated by NMR or X-ray | 86 |
| Average RNA length | 40 |
| Min RNA length | 12 |
| Max length | 545 |
| Average number of stems | 2 |
| Average number of base pairs in stems | 12 |

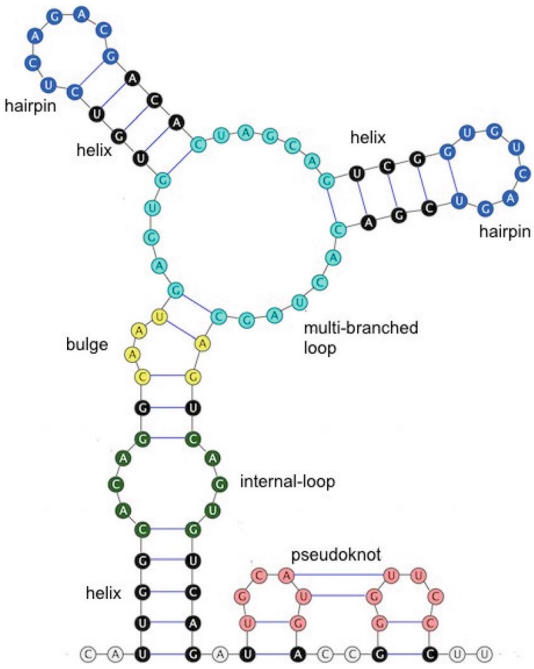


**Fig. 8** Examples of common structural characteristics (Mamuye et al. 2016)

Change at 37° C. Detailed test data along with Python code and Qfold executables are available on GitHub for download and testing using Windows machines (Verma 2020).

### 4.1 Problem characteristics

RNA STRAND problem characteristics vary widely and summary data is provided in Table 1. Figure 8 taken from Mamuye et al. (2016) provides examples of common structural RNA characteristics and Table 2 shows the percentage of structural characteristics in the problems tested. For example 37% of the RNA in the test set have pseudoknots in their benchmark descriptions, which adversely affects comparison measures for these results because Qfold does not currently implement the generation of pseudoknots.

**Table 2** Structural characteristic percentages in problems tested

| | % Includes | % Excludes |
|---|---|---|
| Pseudoknots | 37 | 63 |
| Non-canonical | 36 | 64 |
| Multi-branch loops | 66 | 34 |
| Internal loops | 56 | 44 |
| Hairpins | 59 | 41 |

## 4.2 Results

We measured the performance of Qfold using 431 of the RNAs found in the RNA STRAND database. To calculate measures of comparisons to the benchmarks reported we tallied True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) counts to calculate F-scores and Matthew's Correlation Coefficients (MCC). An F-score can be interpreted as the percent of the structure that is correctly predicted. It is calculated from Precision and Recall: F-score = Precision * Recall/(Precision + Recall), where Precision = TP/(TP + FP) and Recall = TP/(TP + FN). F-score is known as the harmonic mean of Precision and Recall and is a common measure in classification, however it does not take into account True Negatives. MCC combines TN, TP, FN, FP and is widely used in machine learning and bioinformatics. MCC varies between −1 to +1 where +1 is perfect agreement, 0 is same as random, and −1 is no agreement between binary classifications. F-score ranges from 0 to 1 wherein 1 represents the best value having perfect precision and recall.

A comparison of MCC and F-score over RNA length is shown in Fig. 9. The graphs illustrate a gap between MCC and F-score that widens as the length of the primary RNA structure increases, indicating that larger RNA have more ending

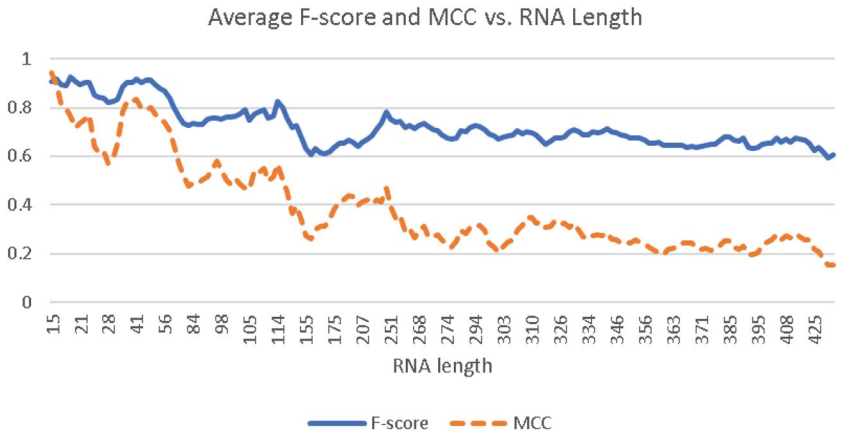


**Fig. 9** Comparison of Qfold F-score and MCC to benchmark data as RNA length increases

state possibilities, that may have low MFE but do not correlate well with the single benchmark structure.

Precision and Recall are also commonly used comparison measures. Precision measures the percent True Positives out of all predicted positives and is meaningful when the emphasis is on avoiding false positives, such as saying a base is paired when it is not paired in the benchmark. Recall measures the percent True Positives out of actual positives and is meaningful when the emphasis is on avoiding false negatives, such as saying a base is not paired when it is paired in the benchmark. As RNA length increases, these measures tend to converge to just above 60% (see Fig. 10). These F-score, Precision and Recall results align with those of other approaches. For example, Chen et al. (2020) reported an average F-score range from 0.4 to 0.6 for the six prediction methods they used for comparison.

Because pseudoknot prediction greatly increases complexity and is associated with tertiary structure prediction, many RNA folding prediction programs do not include the capability to predict pseudoknots. Figure 11 shows the distribution of MCC according to accepted categorizations of MCC values for RNA *without* pseudoknots. The average MCC for the test set of all 268 RNA without pseudoknots was 58% (a strong correlation between Qfold results and the benchmarks) and the percent of moderate to very strong correlations was 78%. Including the RNA with pseudoknots, the average MCC was 48% and 63% of the Qfold models showed moderate to very strong correlation to the benchmark secondary structures.

While Qfold optimizes RNA folding using available free energy measurements between nested base pairs, it also relies on bonuses in the quadratic term to reward stacking variables in order to create stems and these values are not experimentally determined. Because of this, we investigated the correlation between

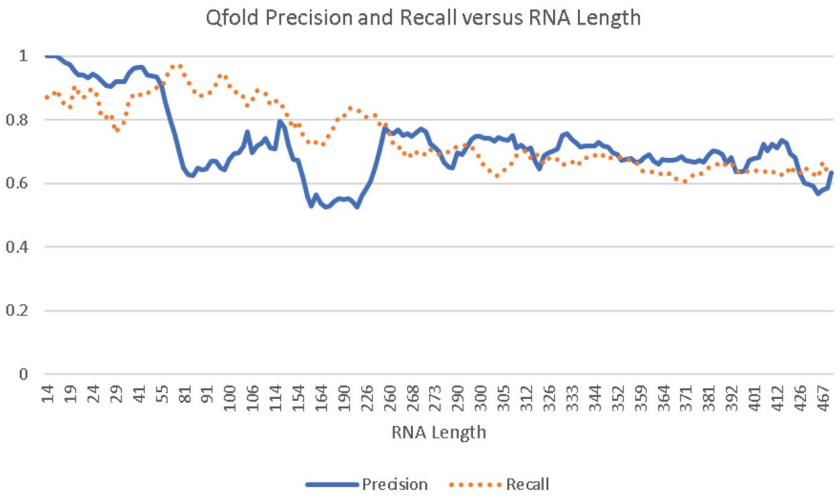


**Fig. 10** Qfold Precision and Recall metrics against benchmark data as RNA length increases

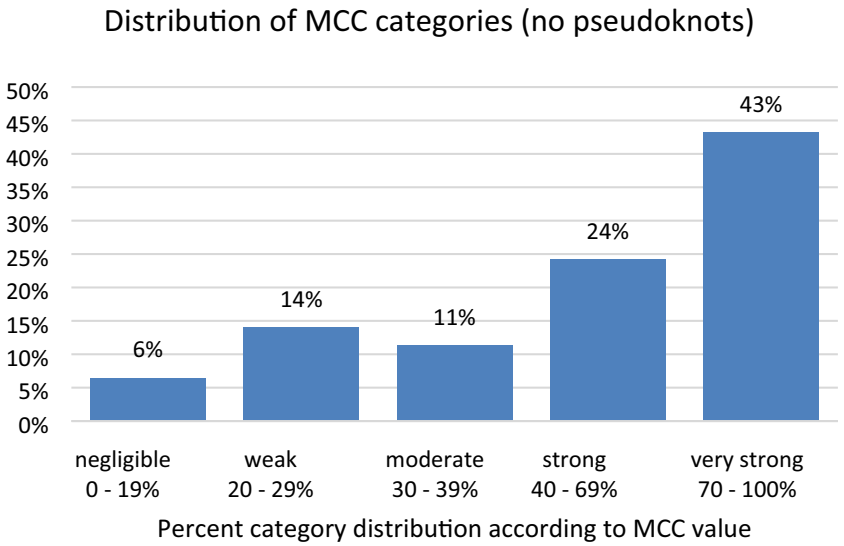


**Fig. 11** Correlation categorization to benchmark secondary structures w/o pseudoknots of 268 RNA

Qfold objective function value (QMFE) and MFE and found a strong positive correlation of R = 62%. We also found a strong correlation of 64% between QMFE and MCC. Thus QMFE is a good proxy for both MFE and MCC.

The number of variables generated for the QUBO is a key concern since problems are generally more difficult to solve as they grow. The effect of RNA length on QUBO size is illustrated in Fig. 12 and shows a quadratic, or $O(n^2)$, relationship. RNAs with over 1000 bases were solved but the QUBO size approached 30,000 variables, which is the limit for Qfold data structures using 64 bit addressing.

We also investigate the correlation coefficient R between structural characteristics and F-score and MCC as those structural characteristics increase. One might ask,

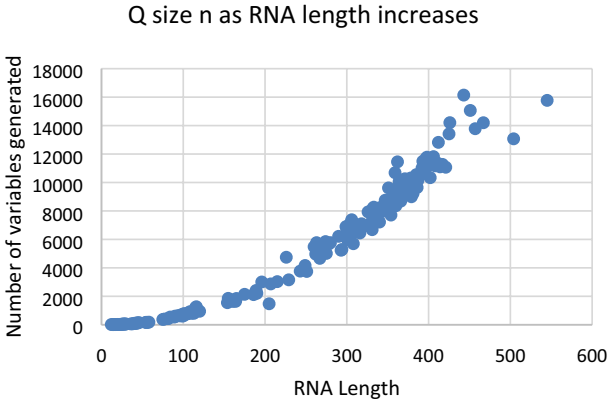


**Fig. 12** QUBO size increases $O(n^2)$ with RNA length

is MCC positively correlated with increases in the number of pseudoknots in the benchmarks? Table 3 shows that as the number of pseudoknots increases F-score and MCC decrease, and in general, as the structural complexity of RNA increases, the similarity to the benchmark secondary structure decreases.

Experimentally verified structures are examined in a stable in vitro environment although structures formed in situ may be differ due to factors not present in the lab. Furthermore, most of the benchmarks use Comparative Sequence Analysis which is an in silico approach—sssnot an experimentally verified in vitro benchmark. Only 20% of the benchmarks are experimentally verified and for this group of RNA the average MCC was 71%, while the average MCC for the Comparative Sequence Analysis group was 42%. The asterisked entries in Table 3 illustrate additional differences. For example, for those RNA whose structure has been experimentally verified, RNA length had no correlation to F-score.

### 4.3 Effect of parameters and weights

Various parameters impact Qfold results. The main parameters in Q_generation (S, min_d, weights) are min_d (minimum distance between base pairs), which affects the feasible base pairs generated, and the weights (refer to Fig. 3) used to create the coefficients in the Q matrix. The minimum distance between base pairs affects the number of feasible variables generated. A smaller minimum distance implies more variables generated and more possible structural characteristics. The interaction weights used in the objective function guide the progression of solutions towards an optimal minimum free energy and a small change in the weights may have large changes in the structure as described in Sect. 4.3.2.

**Table 3** Correlation coefficient R values between structural characteristic growth and F-score and MCC

| Morphology characteristics | Averages | |
|---|---|---|
| | MCC | F |
| Pseudoknots | −0.98 | −0.95 |
| Bulge loops** | −0.86 | −0.24 |
| Hairpins | −0.83 | −0.81 |
| Multi-loops | −0.80 | −0.86 |
| Bulge loops* | −0.71 | −0.50 |
| Number base pairs in stem | −0.55 | −0.43 |
| RNA length* | −0.54 | −0.42 |
| Max stem length* | −0.50 | −0.51 |
| Number of stems | −0.48 | −0.32 |
| Max stem length** | −0.41 | −0.09 |
| RNA length** | −0.36 | 0 |
| Internal loops | −0.27 | −0.10 |

*Verified VIA comparative sequence analysis

**Verified via X-ray or NMR

### 4.3.1 Effect of min_d

The minimum distance parameter defines the number of unpaired bases allowed in a hairpin loop peripherally affects other structures such as bulges and inner loops and thus may affect both the final solution as well as sub-optimal ones. In accordance with other approaches, the default value used was four. In Fig. 13 (top), for molecule PDB_00317 (length = 28 bases), min_d = 4 and this produces the progression of solutions with increases in pairing and stem length and subsequent decreases in free energy. The progression of solutions in Fig. 12 (bottom) uses min_d = 3. Although both settings arrive at the same ending solution and MFE, and these correspond well with the RNA STRAND benchmark, the paths taken are different, generating different sub-optimal solutions and illustrating that min_d can affect solution progression.

### 4.3.2 Effect of base pair weights

The weights used to quantify interactions between nested base pairs in the Q matrix can make a large difference in the results obtained. For Qfold, we implemented the 36 weights used by and Turner and Mathews (2009) and implemented in ViennaRNA. However, the 36 weights used by Kelly and Didulo (2018) at Kelly Bioinformatics (Kelley 2020) differ on average by 11% from those of Turner, with the largest being a 140% difference (1.3 versus −0.5 kcal/mol) in the value of G-U: U-G wobble pairs. The RNA structure PDB_00072 (length = 545 bases) illustrates the effect of these changes. Figure 14(a) shows the secondary structure verified by X-ray crystallography and 14(b) shows the morphology using weights provided by Kelly, while (c) is the Qfold result using weights from ViennaRNA, and (d) is the result of ViennaRNA. No other parameters were changed. The shapes of the two Qfold structures are noticeably different, with the Kelly weights providing a better match to the X-ray verified structure.

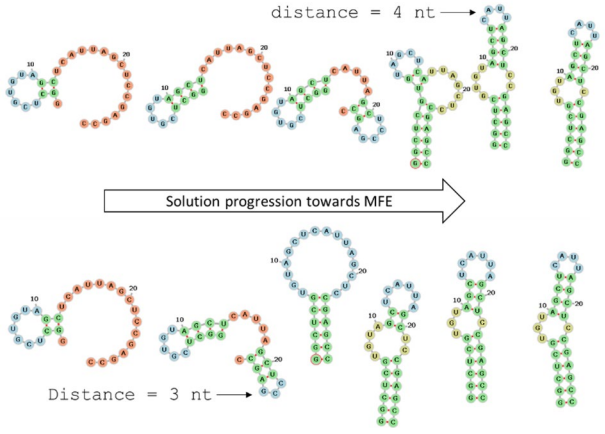


**Fig. 13** Effect of parameter min_d on sub-optimal solutions generated for PDB_00317

a. X-ray b. *Qfold* Kelly weights c. *Qfold* Turner weights d. ViennaRNA solution

**Fig. 14** Morphology changes due to differences in interaction weights

### 4.4 Algorithm component discussion

The intent of this investigation was to show that a relatively simple QUBO model for the RNA folding problem was a reasonable one and that using a generic QUBO solver, as opposed to a customized for the problem, would help demonstrate that the model approach was indeed appropriate. Our definition of a generic QUBO solver was one with three basic elements: efficient 1-flips, methods to handle local optima and proven solution improvement techniques. Customized solution approaches to leverage known structural motifs and manage pseudoknots as well as large problem instances are topics for future research.

To address questions regarding the impact of algorithmic options in the generic solver, Table 4 summarizes the average effects of using Path Relinking (PR) only, PR and Strategic Oscillation (PR + SO) and PR and Backtracking (PR + BT) with column A being the method discussed in Sect. 3.1.2 that incorporates all three options. The results use the same test set as described in Table 1 and indicate that on average method A was faster to the same solution quality. However, the Path Relinking only option appears the better approach for RNA with over 250 bases.

An anecdotal and empirical time complexity analysis indicates the majority of time spent is in the search for local optima while less than 5% of the time was spent in path relinking, backtracking and strategic oscillation phases. These are only triggered after a local has been discovered, so improvements to finding local optima are warranted and future research using more sophisticated solution techniques such as meta-analysis to dynamically adjust backtracking and strategic oscillation

**Table 4** Time and MCC comparisons for alternative implementations of Qfold

| # Bases | Time to solution (s) | | | | MCC (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | PR only | PR + SO | PR + BT | A | PR only | PR + SO | PR + BT |
| 1–50 | 46 | 56 | 57 | 58 | 79 | 79 | 79 | 79 |
| 51–100 | 29 | 43 | 44 | 43 | 70 | 70 | 68 | 69 |
| 101–150 | 28 | 33 | 42 | 47 | 51 | 53 | 51 | 54 |
| 151–250 | 30 | 60 | 60 | 60 | 28 | 23 | 32 | 22 |
| > 250 | 55 | 52 | 59 | 58 | 30 | 35 | 35 | 36 |
| Averages | 38 | 49 | 52 | 53 | 52 | 52 | 53 | 52 |

parameters, as well as improved path relinking via a managed pool of diverse elite solutions along with using relinking during the greedy search phase may be fruitful for improving time to solution and MCC.

## 5 Conclusions

We introduce a Quadratic Unconstrained Binary Optimization (QUBO) model for RNA secondary structure prediction and to the best of our knowledge, no other quadratic binary RNA secondary structure prediction model has been reported in the literature. We present three models in which the binary variables and their quadratic interactions support increasing RNA stem length and explore in detail the model that promotes the formation of long RNA stems in order to reduce free energy. Results indicate our QMFE objective function is strongly correlated to the standard MFE measure used by other RNA folding programs. We describe, implement and test a hybrid metaheuristic for solving QUBO problems and demonstrate results that strongly correlate with RNA benchmarks. The effects of parameters such as minimum distance between base pairs and energy weights on solution progression is presented.

The QUBO model tested in this paper uses a single six by six table of 36 base pair interaction weights, which is only a small subset of the available RNA base interaction data. Other approaches pull data from thirty-six $16\times16$ tables of energies associated with all possible $2\times2$ interior loops in a stem and include details such as differences in energy associated with base position and loop closing versus non-closing base pairs. Therefore, incorporating these additional details into the QUBO model, along with defining binary variables representing structures more complex than stems (motifs) are promising areas for future research. Customizing the search process to avoid, or to seek out, certain structural motifs and adjusting penalty magnitudes to create soft constraints on crossing pairs, hence allowing pseudoknots, is also promising. As RNA folding problems and the QUBO they generate become larger, more efficient and accurate methods of structural prediction will be needed.

## References

Andronescu, M., Bereg, V., Hoos, H.H., Condon, A.: RNA STRAND: the RNA secondary structure and statistical analysis database. BMC Bioinform. **9**(1), 340 (2008)

Barahona, F., Grotschel, M., Junger, M., Reainelt, G.: An application of combinatorial optimization to statistical physics and circuit layout design. Oper. Res. **36**(3), 493–513 (1988)

Beasley, J.E.: OR-Library: distributing test problems via electronic mail. J. Oper. Res. Soc. **41**(11), 1069–1072 (1990)

Boothby, T.K.A.D., Roy, A.: Fast clique minor generation in Chimera qubit connectivity graphs. Quantum Inf. Process. **15**(1), 495–508 (2016)

Chen, X. et al.: RNA secondary structure prediction by learning unrolled algorithms. In: International Conference of Learning Presentations (2020)

Choi, V.: Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. Quantum Inf. Process. **7**, 193–201 (2008)

D-Wave Systems: D-Wave (2020). https://www.dwavesys.com/

Fallmann, J., et al.: Recent advances in RNA folding. J. Biotechnol. **261**, 97–104 (2017)
Findeiss, S., et al.: In silico design of ligand triggered RNA switches. Methods **143**, 90–101 (2018)
Forrester, R., Greenberg, H.: Quadratic binary programming models in computational biology. Algorithmic Oper. Res. **3**(2), 110–129 (2008)
Fujitsu: Digital Annealer—Quantum Computing Technology, Available Today (2020). https://www.fujitsu.com/global/services/business-services/digital-annealer/
Gardner, P., Giegerich, R.: A comprehensive comparison of comparative RNA structure prediction approaches. Bioinformatics **5**, 140 (2004)
Glover, F.: Exploiting Local Optimality in Metaheuristic Search (2020). https://arxiv.org/ftp/arxiv/papers/2010/2010.05394.pdf
Glover, F., Alidaee, B., Rego, C., Kochenberger, G.: One-pass heuristics for large-scale unconstrained binary quadratic problems. Eur. J. Oper. Res. **13**(2), 272–287 (2002)
Glover, F., Kochenberger, G., Du, Y.: Quantum bridge analytics I: a tutorial on formulating and using QUBO models. 4OR Q. J. Oper. Res. **17**, 335–371 (2019)
Glover, F., Lewis, M., Kochenberger, G.: Logical and inequality implications for reducing the size and difficulty of quadratic unconstrained binary optimization problems. Eur. J. Oper. Res. **265**(3), 829–842 (2018)
Gusfield, D.: Chapter 6 The RNA-folding problem. In: Integer Linear Programming in Computational and Systems Biology: An Entry-Level Text and Course. Cambridge University Press, New York (2019)
Hammer, P., Rudeanu, S.: Boolean Methods in Operations Research and Related Areas. Sprnger, Berlin (1968)
Huang, L., et al.: LinearFold: linear-time approximate RNA folding by 5'-to-3'dynamic programming and beam search. Bioinformatics **35**, 295–304 (2019)
ILOG, C.I.: V12 User's Manual for CPLEX (2019)
Kelley, S.: Kelly Bioinformatics (2020). https://www.kelleybioinfo.org/algorithms/default.php?o=3#
Kelly, S., Didulo, D.: Computational Biology: A Hypertextbook, 1st edn. ASM Press, Washington (2018)
Kerpediev, P., Hammer, S., Hofacker, I.: Forna (force-directed RNA): simple and effective online RNA secondary structure diagrams. Bioinformatics **31**(20), 3377–3379 (2015)
Kochenberger, G., Glover, F., Alidaee, B., Rego, C.: A unified modeling and solution framework for combinatorial optimization problems. OR Spectrum **26**(3), 237–250 (2004)
Kochenberger, G., et al.: The unconstrained binary quadratic programming problem: a survey. J. Comb. Optim. **28**, 58–81 (2014)
Laguna, M., Glover, F.: Integrating target analysis and tabu search for improved scheduling systems. Expert Syst. Appl. **6**, 287–292 (1993)
Lewis, M., Kochenberger, G.: Probabilistic multistart with path relinking for solving the unconstrained binary quadratic problem. Int J Oper Res **26**(1), 13–33 (2016)
Lucas, A.: Ising formulations of many NP problems. Front. Phys. **2**, 5 (2014)
Mamuye, A., Merelli, E., Tesei, L.: A graph grammar for modelling RNA folding. Electr. Proc. Theor. Comput. Sci. **231**, 31–41 (2016)
Mathews, D.: Free Energy and Enthalpy Change Parameters (2020). https://rna.urmc.rochester.edu/NNDB/turner04/index.html
Mathews, D.H.: How to benchmark RNA secondary structure prediction accuracy. Methods **162**, 60–67 (2019)
Mauri, G.R., Lorena, L.A.N.: A column generation approach for the unconstrained binary quadratic programming problem. Eur. J. Oper. Res. **217**, 69–74 (2012)
Meta-Analytics: Alpha-QUBO: Optimization Technology for the modern age (2020). http://meta-analytics.net/Home/AlphaQUBO. Accessed 2020
Palubeckis, G.: Iterated tabu search for the unconstrained binary quadratic optimization problem. Informatica **17**(2), 279–296 (2006)
Pardalos, P., Jha, S.: Complexity of uniqueness and local search in quadratic 0–1 programming. Oper Res Lett **11**(2), 119–123 (1992)
Pardalos, P.M., Rodgers, G.P.: Computational aspects of a branch and bound algorithm for quadratic zero-one programming. Computing **45**(2), 131–144 (1990a)
RNA STRAND Database: RNA STRAND (2008). http://www.rnasoft.ca/strand/. Accessed 2020 June.
Saad, S., Backofen, R., Ponty, Y.: Impact of the Energy Model on the Complexity of RNA Folding with Pseudoknots. In: Karkkainen, J., Stoye, J. (eds.) Combinatorial Pattern Matching, pp. 321–333. Springer, Berlin (2012)

Shi, S., et al.: Prediction of the RNA secondary structure using a multi-population assisted quantum genetic algorithm. Hum Heredity **84**, 1–8 (2019)
Singh, J., Hanson, J., Paliwal, K., Zhou, Y.: RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. Nat. Commun. **12**, 1–13 (2019)
Turner, D., Mathews, D.: NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. Nucleic Acids Res. **38**, 280–282 (2009)
Verma, A.: Qfold (2020). https://github.com/amitverma1509/Qfold. Accessed 25 June 2020
Vienna RNA Web Service: RNA Secondary Structure Visualization Using a Force Directed Graph Layout (2020). http://rna.tbi.univie.ac.at/forna/. Accessed 2020
Wang, Y., Lu, Z., Glover, F., Hao, J.: Path relinking for unconstrained binary quadratic programming. Eur. J. Oper. Res. **223**(3), 595–604 (2012)
Watson, J.D., Crick, F.H.C.: A structure for deoxyribose nucleic acid. Nature **171**, 737–738 (1953)
Yan, Z., Hamilton, W., Blanchette, M.: Graph neural representational learning of RNA secondary structures for predicting RNA-protein interactions (2020). https://doi.org/10.1101/2020.02.11.931030
Zhang, H., et al.: A new method of RNA secondary structure prediction based on convolutional neural network and dynamic programming. Front. Genet. **10**, 467 (2019)
Zuker, M., Stiegler, P.: Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res. **9**, 133–148 (1981)