Inverse folding of RNA II

Rune B. Lyngsø and James W. J. Anderson

Abstract

In the Oxford Summer School in Computational Biology 2011, one of the projects was aimed at creating an improved method for inverse RNA folding based on a genetic algorithm (GA) approach. This turned out to be sufficiently successful and inspiring, that we will extend on it in this year’s summer school. The inverse RNA folding problem consists of finding an RNA sequence that, as a molecule, folds into a target structure, or structures. We will focus exclusively on what is known as the secondary structure of the molecule, which is the set of base pair interactions formed in the structure. The secondary structure can be inferred from the sequence with reasonable precision, with several paradigms available. This allows the inverse folding problem to be addressed completely in silico (i.e. computationally), using predicted structures as a proxy for the true structure of a designed sequence. The aim of this follow-up project is to address at least some of the following points:

1. enhance the existing GA with functionality for multi-objective fitness and objective functions
2. extend the method to utilise more folding paradigms to create more robust designs
3. extend the software with a graphical user interface and parallelism
4. improve the sequence initialisation and search methods to be able to design perfect sequences for more targets
5. extend the target specification to include structures with pseudo-knots, nucleotide constraints for specific positions as well as overall distributions, less specific targets with regions where any structure is allowed or even as more abstract shapes, and possibly even two or more molecules forming complex structures
6. improve complexity results for the inverse RNA folding problem

It is not realistic to assume that all these points can be addressed in full, or possibly even at all. The list should work as a source of inspiration, from which you can choose the most relevant and interesting elements to work on. However, the ordering does reflect our a priori prioritisation.

This project description first describes the basic background for the inverse folding problem, including a brief summary of our method. It then outlines the points for extensions in more details, also discussing the importance of each proposed extension.
1 Background

The advent of nano-sciences, where we want to design small entities with specific structure or behaviour, has already spurred an interest in using nucleic acid sequence molecules as the fundamental building block, with some rather curious examples (see e.g. [22, 2]). It is only natural that DNA and in particular RNA molecules have been considered for nano-scale design as:

- numerous motifs with specific functionalities are known from biology
- large scale production can be carried out by normal replication of simple organisms
- we already possess an extensive understanding of structure formation for nucleic acid sequences, allowing relatively good predictions of both secondary and tertiary structure purely by computational means

The aim of the basic version of the inverse folding problem for RNA is, given a target structure like e.g. the one depicted in Fig. 1, find a sequence that folds into this structure.

Figure 1: An example target RNA structure. The sequence backbone follows the straight lines with circles representing the nucleotides. Nucleotides connected by a zigzag line should form base pair interactions in the stable structure for the designed sequence.

In this project we will exclusively focus on the secondary structure. The main driving force behind RNA structure formation is the creation of base pairs similar to the ones observed in the DNA double helical structure. In RNA thymine is replaced with uracil, so the Watson-Crick base pairs of RNA are between C and G, and between A and U. Additionally the so-called wobble base pair between G and U is commonly observed, and these six (when accounting for ordering) types of base pairs are known as the canonical base pairs of RNA. The main difference between DNA and RNA is that where the DNA helix is formed between complementary strands, an RNA molecule consists of just a
single strand, or sequence, and the helices are formed locally between different parts of the sequence. The secondary structure of an RNA molecule is the set of base pair interactions observed in the full three dimensional structure.

In lieu of costly experiments to determine the true structure, the secondary structure predicted by standard methods is usually used to determine the fold of proposed sequences. RNA secondary structure prediction is one of the classical problems in computational biology [9]. For decades it has been applied to make inferences about the structure of sequenced RNA, and more recently as an integral part of non-coding RNA gene finding [28, 20]. One standard method is based on a model assigning free energies to secondary structures [27, 17]. Thermodynamics then states that the most stable structure is the one with lowest free energy, and this is usually taken to be the true structure. This model has the further advantage of defining a (Boltzmann) distribution over structures from which e.g. individual base pair probabilities can be estimated [18]. A closely related approach uses stochastic context-free grammars [8], allowing similar inferences to be made. This has more recently been extended to conditional log-linear models [7], and methods based on assembling global structures from small, local motifs [19].

A simple, but time consuming, approach to find a sequence folding to a particular structure would be to enumerate all the $O(4^n)$ possible sequences of the length $n$ of the desired target structure, checking whether any has a predicted structure matching the target. There is no known efficient method for computational design of RNA molecules folding to a specific structure, and some evidence that no such method exists [23]. A number of heuristic methods have been proposed [12, 10, 8, 7, 5, 11, 29, 25, 20, 4], all with a some sort of a random walk local search where one or two nucleotides are changed in each step as a core constituent part. Some include other key parts, like improved sequence initialisation [5] or hierarchical decomposition of the search [8, 7]. However, only the method of [25, 26] extends the local search – that is essentially equivalent to a GA mutation operation – with a cross-over operation to provide a full GA approach. This latter method is however somewhat hamstrung by a limited familiarity with the application area. Given how ideally the problem is suited for a GA approach, with a decomposability of the problem also utilised by the hierarchical approach of [3, 1], this is rather surprising.

2 Pilot

As the focus of this project is mainly to extend on our existing method [16] for inverse RNA folding, a good way to work yourself into it will be to implement some fairly straightforward, nice-to-have features. These will include constraints on nucleotides in specific positions and overall nucleotide content, as well as interfacing to an alternative folding paradigm. This should be integrated into the existing Python implementation Frnakenstein, available from www.stats.ox.ac.uk/research/genome/software/frnakenstein. More specifically, the features we would like to add are:
• In addition to folding into the specified target structure, sometimes it may also be important that the designed sequence has specific nucleotides in certain positions to function as intended. The current method should thus be extended with a means for specifying position specific constraints, e.g. as a sequence over the extended RNA alphabet, cf. [www.bioperl.org/wiki/BioPerl_Alphabets](http://www.bioperl.org/wiki/BioPerl_Alphabets). The final design should abide by these constraints, so we will aim to maintain this property throughout the GA evolution. This means that the constraints will affect sequence initialisation, as well as mutations.

• A more global requirement on sequence contents could be requirements on overall nucleotide frequencies. The current implementation already supports drawing new nucleotides from specified distributions, but selection in the GA may result in deviations from these distributions for the final sequence. A simple approach to better ensure a desired frequency distribution would be to add an element to sequence fitness capturing deviation from the target frequencies.

• One aim of the full project is to require sequence to have a predicted structure matching the target structure under several folding paradigms. Currently, only thermodynamic folding [9, 30] is supported, through the interface to RNAfold (available from [www.tbi.univie.ac.at/~ivo/RNA/](http://www.tbi.univie.ac.at/~ivo/RNA/)) in the mfe.py module. The main alternative paradigm is based on stochastic context-free grammars [8] (SCFGs). Reimplementing the interface using e.g. PPfold [24], available from [www.daimi.au.dk/~compbio/pfold/downloads.html](http://www.daimi.au.dk/~compbio/pfold/downloads.html), will allow the inverse folding to be carried out using SCFG based structure predictions.

Ideally, this pilot should be completed within the first two weeks of the summer school.

3 Project

The aim of the main project does not differ from the pilot described in Sec. 2 in nature. We have several ideas of how the existing Frnakenstein version can be improved, which we would like to extend the existing software with. However, the tasks in this section will tend to be more complex. New ideas and insights usually result from working with a problem, so the following description should not be taken as a fixed to-do list, but rather as an outline guide to how we from our current base could see the project develop. The first two elements in the following list of extensions are our key priorities for this project, with the remaining elements more open selection based on your personal preferences.

3.1 Multi-dimensional fitness function

The one existing GA approach to the inverse folding problem [25, 26] uses a two-dimensional fitness function, consisting of distance from predicted structure to
target structure and free energy of sequence on target structure, when selecting individuals to carry over to the next generation. The current implementation of Frnakenstein does allow different criteria to be combined, but only by reducing them to a single fitness score e.g. by taking a weighted sum. A key extension will be to extend the software to support the same kind of multi-objective selection that is used in Modena.

Apart from necessary changes to core parts of the software, regarding how fitnesses are computed and represented, the selection method also needs to be changed. When fitness is reduced to a single number, there is a total ordering on all individuals. This allows obvious choices for selection, either deterministically choosing the best individuals or some kind of randomised scheme choosing individuals according to their fitness score. With a fitness composed of two or more scores, there may be no one individual beating all the others on all scores. The natural choice becomes one of choosing Pareto optimal sets, as described in [25]: we want to choose the subset of individuals for which there does not exist any other individual which is better on all scores (i.e. individuals with scores not dominated by the scores of any other individual, for those of you who have not repressed all memories of the interview). This can be done either using a simple method of comparing all against all, or using more efficient algorithms, see e.g. [13] and literature on skyline queries. It is unlikely that this selection step is going to be time critical, so our suggestion would be to implement the simple method unless you have an interest in understanding the more complex and efficient approaches.

3.2 Robust designs based on multiple paradigms

The different structure prediction paradigms have comparable performance, which unfortunately is not perfect. Moreover, under the SCFG paradigm, different grammars can be used, also exhibiting similar performance. However, even if overall performance is similar, there is a large variation in which structures each paradigm and grammar predicts well. For a more robust design, it is thus natural to search for sequences folding into the target structure under several different prediction methods. With the grammar paradigm interface of the pilot in Sec. 2 and multi-dimensional fitness functions of Sec. 3.1 in place, it should be fairly straightforward to extend Frnakenstein to use the distance to the target structure for multiple predictions in the fitness function selection is based on.

3.3 Pseudoknot design

The standard approach to RNA structure prediction, under both the thermodynamic and SCFG paradigms, does not allow prediction of structures with pseudoknots. A pseudoknot occurs when two base pairs overlap, i.e. if positions $i$ and $j$ form a base pair and positions $i'$ and $j'$ form a base pair and $i < i' < j < j'$. The structure prediction becomes much harder when allowing pseudoknots, but some restricted and/or heuristic methods exist [14, 21].
A few other methods \[11, 26\] methods allow design of sequences folding into structures containing pseudoknots. As was (hopefully) demonstrated in the pilot, it is fairly easy to change the prediction method used for assessing candidate solutions. However, to fully support design of structures containing pseudoknots, the cross-over operation in \textit{Frnakenstein} would have to be updated. Currently it to some extent relies on the hierarchical nature of unknotted structures, and for certain pseudoknotted structures this reliance may result in too small a set of possible cross-over points.

### 3.4 Parallelism

By far the main time consuming step of \textit{Frnakenstein} is the external call to the structure prediction method. With more knowledge of the internal operation of the structure prediction method, it could be possible to reduce computation time by reusing parts of previous computations. However, this would be a complex task and would make \textit{Frnakenstein} intrinsically tied to one or two specific prediction methods, making the extension in Sec. \[3.2\] much harder to maintain and incorporation of future improved methods more difficult.

This leaves parallelising the structure predictions the most feasible route to reduced running time, in particular given the ubiquity of multi-core processors. The current implementation has been tailored to be easily extended to perform the external structure prediction calls in parallel, but the actual functionality still needs to be implemented.

### 3.5 Graphical user interface

In the Good Old Days, people were happy with a command line interface, providing a plethora of options for specialising the behaviour. Sadly, those days are gone, so software no either has to provide some unique and really useful functionality or come with a graphical user interface (GUI) to be successful. The current version of \textit{Frnakenstein} only provides a command line interface, so a useful addition would be to add a GUI allowing targets to be loaded from files, interfaces for setting all the choices defining the GA, and providing updates on progress and current state during the run.

### 3.6 Improved search methods

There are some structural motifs, like isolated base pairs and short stems, that it is very difficult to design sequences for. Rather than searching for solutions to such motifs by the random mutations and recombinations of the GA, it may improve the overall search if a library of such motifs and their solutions – or information that no exact solution exists, together with closest matching solutions – was developed and used, e.g. by allowing larger scale mutations where the sequence for an occurrence of one of these motifs is drawn from the library.
In the extreme case, it would be ideal to have a solution for the entire target structure. As already mentioned, and further discussed in Sec. 3.9, there are indications that an efficient algorithmic solution is not possible. However, one could still try to develop more rational approaches to design sequences that are good starting points for the GA search, rather than the current approach of starting with a sequence compatible with the target structure. For example, [5] finds the sequence resulting in the target structure having the lowest free energy, but this may result in other structures also having extremely low free energy. Hence, the target structure may only be stable compared to the unpaired structure but not compared to other structures. The method of [3, 1] does attempt to build an initial sequence avoiding repetitions of patterns and their reverse complements, but this appears to be done in an ad hoc manner.

A similar problem of avoiding strongly structured elements in mRNA has previously been considered in [6]. Here the aim is to make sure that no structure has low energy. The paper mentions de Bruijn sequences, but proceeds to develop heuristics based on alternative methods. Order $k$ de Bruijn sequences are sequences that contain each length $k$ string exactly once, e.g. an order 2 binary de Bruijn sequence could be 00110 or 01100. Order $k$ de Bruijn sequences can easily be constructed from Hamiltonian paths in finite automata, see e.g. en.wikipedia.org/wiki/De_Bruijn_sequence. Here we are in a somewhat more complicated situation, as we also want to have base pairing segments be reverse complements of each other and otherwise avoid reverse complements. Still, de Bruijn sequences and other similar concepts could be explored for more rational designs of the initial sequence.

3.7 Flexible targets

In many applications of the inverse RNA folding problem, small deviations from the target structure may not be of significant importance. There may even be cases where only a core motif is relevant, while the constraints on the remaining part of the molecule are minimal. In [4] this is addressed to some extent, but mostly with a view to expediency, allowing their method to fail to produce a sequence folding into the target structure as long as it folds into a structure with the same overall shape.

A nice extension to Frankenstein would be support more flexible targets to be specified, e.g. allowing some regions to be unconstrained, other regions to only have shape constraints, and core motifs to be fully constrained. Allowing unconstrained regions should be fairly straightforward within the current framework, but also supporting shape constraints and possibly even variation in length of the designed sequence would be a more complex task.

Additionally, RNA secondary structures should ideally be viewed as a distribution over an ensemble of structures, rather than as a single, fixed structure. Structural changes are more often shifts in such distributions than a refolding into a completely different structure. Experimental methods for probing structures will thus often generate probabilities of specific base pairs being present or regions being paired and unpaired, rather than Boolean statements on base
pair absence and presence. Extending Frnakenstein to design sequences with best match to particular Boltzmann distributions, or summaries of Boltzmann distributions like probability of a position forming a base pair, would make it a more powerful method better aligned with current understanding of RNA secondary structure and experimental techniques. Along the same lines, it would also be beneficial having the ability to assign smaller or larger weights to the match in different regions, creating a more flexible framework than the Boolean division into constrained and unconstrained regions suggested above.

3.8 Complex forming targets

Quite often we will be interested in not just a single RNA molecule folding into a particular structure, but two or more molecules that, when they come together, fold into a specific complex, i.e. binding the molecules together in an overall structure. Designing a group of sequences forming a target complex is supported by [29], and a similar feature would be nice to have in Frnakenstein. Furthermore, as methods for predicting interactions between RNA molecules already exist, this should be fairly easy to implement similarly to alternative prediction paradigms. The main effort will be in modifying target representation to reflect the multiple structures.

3.9 Improved complexity results

As alluded to earlier there is some evidence that inverse RNA folding is not an easy problem. In [23] it is proved \( \text{NP} \)-complete to determine if for a given hidden Markov model (HMM) and target path there exists a sequence with the target path as Viterbi path, i.e. most probable path. Stochastic context-free grammars (SCFGs) are generalisations of HMMs. As already mentioned, SCFGs can be used for RNA secondary structure prediction. So it would appear that determining whether there is a sequence that has a given derivation (i.e. secondary structure) as its most probable is hard.

However, the proof in [23] assumes that the HMM is part of the input. When predicting RNA secondary structures we usually use the same SCFG for all sequences. The question is whether it is possible to strengthen the proof to use a fixed HMM, or even better, a standard SCFG for RNA secondary structure prediction or a simple version of the thermodynamic model similar to the ones proposed in [15]. Alternatively, one could explore the possibility of exploiting the fixed nature of the HMM/SCFG to design sequences with given target paths as viterbi paths. This would prove that the result in [24] does not apply to a fixed design problem, like inverse RNA folding, but only when the model we are designing for is part of the input.
References


References marked with ** are key references that you should read in detail, while references marked with * are important references that you should read.