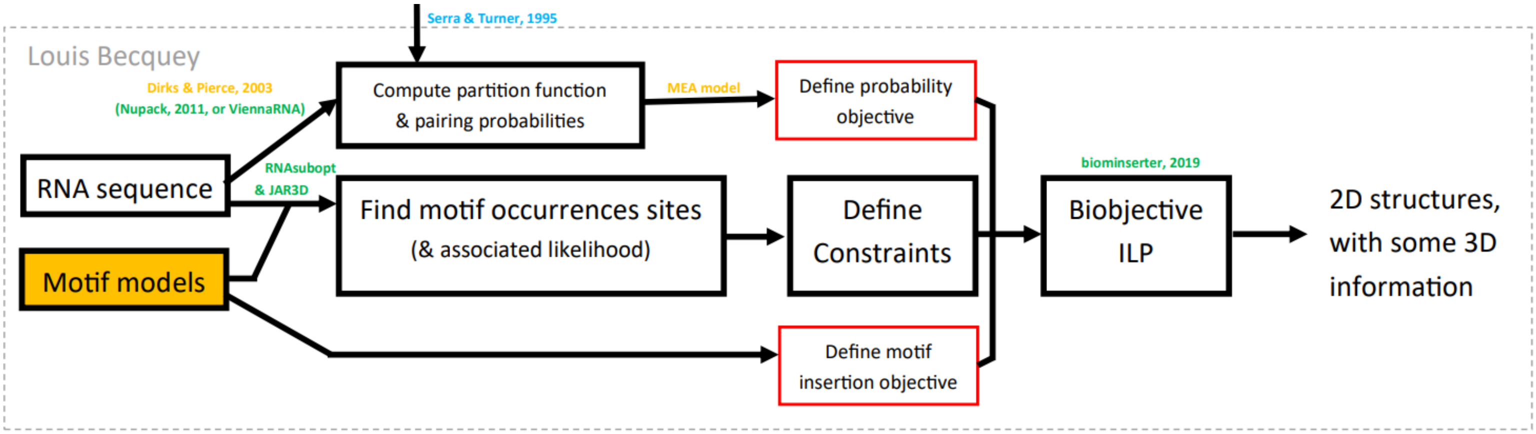
INTERMEDIATE WORK REPORT:

Louis Becquey, February 7, 2019

A biobjective IP framework that uses detected possible RNA modules to predict RNA secondary structures

under the supervision of:
Fariza Tahi, HDR
Eric Angel, HDR
AROBAS, IBISC, Paris-Saclay University

Graphical Abstract


Louis Becquey
Serra & Turner, 1995
Dirks & Pierce, 2003
(Nupack, 2011, or ViennaRNA)
Compute partition function & pairing probabilities
MEA model
Define probability objective
RNA sequence
RNAsubopt & JAR3D
Find motif occurrences sites (& associated likelihood)
Define Constraints
biominserter, 2019
Biobjective ILP
2D structures, with some 3D information
Motif models
Define motif insertion objective

Contents

1 Motivation

We now have quite consistent RNA module databases (RNA3Dmodule [1], the RNA Motif Atlas [2], or CaRNAval [3]). We call a RNA module the combined description of these points:

- A particular base-pairing pattern of canonical & wobble pairing (bi-dimensional information),
- A particular organisation of non-canonical contacts in space (tri-dimensional information).
- A sequence or consensus sequence that we know to adopt that particular base-pairing organisation (sequence information), or at least a probabilistic model to predict if a given sequence will fold according to the module.

Then, a RNA module is direct data about how a sequence can fold in space. Here, we are interested in using them to predict the bi or tri dimensional structure of an RNA sequence from scratch. Then, the algorithm we plan to use is the following:

- Find all possible occurences of known RNA modules in the query sequence by using the sequence probabilistic models of the modules against the query bases,
- Define constraints on the secondary structure imposed by motives if they would be included
- Find a secondary structure that satisfies as much as possible both the expected accuracy of the structure and a criterion of module inclusion, by solving a bi-objective IP problem, using the previous constraints.

2 Computation of basepair probabilities

To compute the probability of a structure to exist in the equilibrium thermodynamical ensemble, we need to estimate the basepair probabilities. This is usually done by a dynamic programming algorithm using an energy parameter model for base stackings in stems and energies of loops. As we are supposed to allow the existence of pseudoknots, we chose the Dirks & Pierce algorithm [4] and its C++ implementation called Nupack.

Discussion of that choice
The Dirks and Pierce algorithm has a complexity of $O(N^5)$ instead of the usual $O(N^3)$ algorithm, and its implementation in libNupack is even not parallel. We may want to use a faster one (like ViennaRNA's one), even if it does not officialy supports pseudoknots. Actually, to form a simple class of pseudoknots, an RNA needs to have a high likelihood to form each of the two independant loops in a non-pseudoknotted form. Then, using a probability matrix computed with an algorithm that does not model pseudoknot may result in similar structures. Our biobjective program would etablish the pseudoknot after all as forming more stems stabilizes the structure and as we do not forbid the pseudoknot.

3 Detection of potential modules in the sequence input

Choice of a module model, choice of a sequence probabilistic model
Depending on the authors and databases, different approaches have been published. A very complete model has been developped by the BGSU team [5] : For each recurrent module of their "3D Motif Atlas", they build a probabilistic model for sequence variability based on a hybrid Stochastic Context-Free Grammar/Markov Random Field (SCFG/MRF) method. Concerning the other databases, a probabilistic model using a bayesian Network can be designed for an RNA module [6] to estimate the probability that the sequence would fold into the module.

As BGSU's model is already implemented in a software called JAR3D that perfectly suits our need, we decided to use BGSU's model of modules and their database. JAR3D finds, given the sequence of an RNA loop with no additional information, the likelihood that it folds in every known 3D module of the RNA Motif Atlas [2].

For each module group of the database, JAR3D estimates the probability that a sequence folds into the 3D pattern, given instances of the module in real RNA 3D structures. The instances might differ in sequence, length of bulged insertions, and other details, but share a common graph of 3D non-canonical interactions. It takes into account crossed contacts, base-ribose and base-phosphate interactions, and triple base-pairs. This is the most complete approach to use to my knowledge.

Detection of the loops in an RNA sequence
As JAR3D scores the sequence of RNA loops, we first need to predict the 2D loops positions in the RNA sequence.

For now, we have been using RNAsubopt from the ViennaRNA package [7] to compute the 10 most probable 2D structures of the RNA and extract the loops that occur in those structures. This computation should be avoided in the future : It requires the computation of the RNA's partition function and basepair probabilities again, which is something we already computed with Nupack. Unfortunately, to avoid computing it twice, we should reimplement ourselves the algorithms instead of using the actual implementations. We have programs that output the basepair matrix but no structures, and programs that output structures but not the basepair matrix, and we don't have a program that can do both.

4 Formulation of the IP problem

This formulation is an improved merge between the IPknot [8] formulation to fold RNAs with pseudoknots (but without "levels") and RNAMoIP's one [9] to include modules. The bi-objective algorithm is inspired from BiokoP [10] and detailed later in section 5.

4.1 Variables

Let n be the number of nucleotides in the query RNA sequence s.
Let M be the set of modules that could be inserted in s (a loop from s has a good JAR3D score against one of the modules).
Let x be a module of M, $\|x\|$ be the number of distinct components of x, and $p(x)$ the associated score of insertion given by JAR3D for that motif inserted at a particular position.
Let $P_{x,i}$ be the position in s where we can insert the ith component of module x.
As the same module model can be inserted several times in s, several different x modules in M may refer to the same theoretical module, but inserted at different positions.
Let $k_{x,i}$ be the size in nucleotides of that ith component of x.
Let y_v^u be the **decision boolean variable** indicating that $s[u]$ and $s[v]$ form a canonical base pairing. According to the standard loop model, we always have $v > u + 3$.
Let C_i^x be the **decision boolean variable** indicating that we do insert the ith component of module x at position $P_{x,i}$.

Note that a base pair y_v^u is possible if and only if $v > u + 3$, and that we do not need to use two variables y_v^u and y_{vu} for the same pair. Then, we have $\sum_{i=4}^{n}(n-i)$ decision variables ($\approx \frac{1}{2}n^2$ decision variables) of the form y_v^u. Regarding the C_i^x, if we have an average insertion of ν motives by RNA sequence, the motives having in average μ components, components that can be inserted in average at π different positions in s, then we need to add, in average, $\nu \times \mu \times \pi$ decision variables C_i^x.

Then, we expect having around $\frac{1}{2}n^2 + \nu\mu\pi$ decision variables.

4.2 Objectives

We have two objectives : Find a structure with correct expected accuracy, and find a structure which includes (large) known modules. Let X be the vector of all our decision variables, we define the following objective functions to maximize:

$$f_{1A}(X) = \sum_{x \in M} (\|x\|)^2 \times C_1^x$$

$$f_{1B}(X) = \sum_{x \in M} p(x) \times C_1^x$$

$$f_{1C}(X) = \sum_{x \in M} \left[\frac{\|x\|}{\log_2(\sum_{i=1}^{\|x\|} k_{x,i})} \times p(x) \times C_1^x \right]$$

$$f_2(X) = \sum_{u<v} p_{uv} \times y_v^u \times I[p_{uv} > \theta], \qquad p_{uv} = \sum_{\sigma \in S(s)} y_v^u . p(\sigma|s)$$

The different f_1 objectives are supposed to maximize the number of inserted motives in s, weighted by their number of components (squared for 1A), or the JAR3D score (B and C). In $1C$, a penalty is added on the number of nucleotides involved in the looped zone (sum of $k_{x,i}$) to avoid long unpaired zones. We don't know yet which one will give the better results. f_2 is supposed to maximise the expected accuracy of the secondary structure. p_{uv} are the base pairing probabilities.

Note that f_{1A} is taken from RNA MoIP [9], to compare performance.

4.3 9 Constraints to bind them all

Constraint to ensure there only is 0 or 1 canonical pairing by nucleotide

$$\sum_{v<u} y_u^v + \sum_{v>u} y_v^u \leq 1 \qquad \forall u \in [\![1, n]\!] \tag{1}$$

Constraints to forbid lonely base pairs

$$y_{v+1}^{u-1} - y_v^u + y_{v-1}^{u+1} \geq 0 \qquad \forall (u, v) \in \{(u, v) \in [\![1, n]\!]^2 \mid u + 3 < v\} \tag{2}$$

A basepair should be accompanied by one of its neighbours, forming a stable structure stabilized by stacking energies. In theory, this might add up to $\frac{1}{2}n^2$ constraints, but in practice, this number is very reasonable as the only decision variables kept are those with probability above a θ threshold. Then, this condition sets to zero "lonely decision variables" who have no neighbour basepair variable allowed.

Constraint to forbid pairings inside a module component

$$(k_{x,i} - 2)\ C_i^x + \sum_{u=P_{x,i}+1}^{P_{x,i}+k_{x,i}-2} \left[\sum_{v>u} y_v^u + \sum_{v<u} y_u^v \right] \leq (k_{x,i} - 2) \qquad \forall x \in M, i \in [\![1, \|x\|]\!] \tag{3}$$

Constraint to forbid component to overlap

$$\sum_{x \in M} \sum_{i=1}^{\|x\|} C_i^x \times I(u \in [P_{x,i}; P_{x,i} + k_{x,i} - 1]) \leq 1 \qquad \forall u \in [\![1, n]\!] \tag{4}$$

$I(u \in [P_{x,i}; P_{x,i} + k_{x,i} - 1])$ is a booleean value depending on the condition's truth. Then, whatever the nucleotide u, it can be part of a module component only once.

Constraints to respect the structure of large motives ($\{x \in M \mid \|x\| \geq 2\}$)

This constraint ensures that none or all the components of a motif are inserted.

$$\sum_{i=2}^{\|x\|} C_i^x = (\|x\| - 1) \times C_1^x \qquad \forall x \in \{x \in M \mid \|x\| \geq 2\} \tag{5}$$

And then, we force base pairs between the end of a component and the beginning of the next one:

$$C_1^x \leq y_{P_{x,\|x\|}+k_{x,\|x\|}-1}^{P_{x,1}} \qquad \forall x \in \{x \in M \mid \|x\| \geq 2\} \tag{6}$$

$$C_j^x \leq y_{P_{x,j+1}}^{P_{x,j}+k_{x,j}-1} \qquad \forall x \in \{x \in M \mid \|x\| \geq 2\}, \forall j \in [\![1, \|x\|[\![\tag{7}$$

Constraint 6 binds the first nucleotide of first component to the last one of the last component. Constraint 7 binds the last nucleotide of component j to the first of component $j+1$.

Constraint to forbid a previously found solution

As several solutions may result in the same values of the two objectives, we can't forbid the algorithm to search twice the same region of the objective landscape. We have to explicitly forbid to find again every found solution. We do it by adding iteratively, for every structure s^* found, the following condition :

$$\sum_{y_v^u \in \{y_v^u | y_v^u = 1 \text{ in } s^*\}} (1 - y_v^u) + \sum_{y_v^u \in \{y_v^u | y_v^u = 0 \text{ in } s^*\}} y_v^u + \sum_{C_i^x \in \{C_i^x | C_i^x = 1 \text{ in } s^*\}} (1 - C_i^x) + \sum_{C_i^x \in \{C_i^x | C_i^x = 0 \text{ in } s^*\}} C_i^x \geq 1 \tag{8}$$

It ensures that at least one of the decision variables differs from s^*.

5 Methods

5.1 Bi-objective algorithm

This is an adaptation of BiokoP's algorithm [10] to gather all the points of the pareto set, removing the k-pareto set part.

We start by solving each objective independantly to have a lower and higher bound on each objective. The two solutions found are considered optimal (higher bound) on one objective, and the worse point of the Pareto set concerning the other objective.

Then, we will iteratively solve the mono-objective problem, but adding as a constraint that the second one has to be included between the bounds. Suppose we decide to iteratively solve objective 1. The found solutions are getting worse and worse concerning objective 1, but better and better concerning objective 2. Every time a solution is found with a better objective 2 value, we update our lower bound to search for solutions with objective 2 above this new value. Note that we use weak inequality constraints, as several solutions may have the same values concerning the two objectives and that we want to include them in the Pareto set.

When no more solutions are discovered, the Pareto has been entirely found.

5.2 Solving the IP problem

We use ILOG CPLEX's [11] concert technology to solve the integer linear programming problem. All our decision variables are booleans, and all our constraints are linear.

5.3 Benchmarking of the module inclusion objective functions

To assess the performance of the objective functions proposed in section 4.2, we need to chose some performance metrics. We will focus on:

- wether the native secondary structure of the proposed RNA sequence exists in the returned solutions (the pareto set),
- the number of solutions returned (size of the Pareto set).

The performance is assessed on the structures taken from the RNA STRAND database [12], after a simple preprocessing to remove pseudobases.

6 Results

6.1 Comparison of the 3 objective functions for motif insertion

It appears that function f_{1A} introduced in section 4.2 is not usable in practice. Actually, on a short RNA sequence (67 nucleotides) but with 4 loop sites that may fold into 30 different known loops, we obtain a combinatorial superposition of near $30^4 = 810000$ structures which have the same f_{1A} and f_2 objectives values, and all belong to the Pareto front. This results is not desired, as the size of the Pareto set is wanted to be small, and as the program fills the machine's memory very quick in practice.

7 Discussion

References

[1] Mahassine Djelloul and Alain Denise. Automated motif extraction and classification in RNA tertiary structures. *RNA*, 14(12):2489–2497, January 2008.

[2] Anton I. Petrov, Craig L. Zirbel, and Neocles B. Leontis. Automated classification of RNA 3d motifs and the RNA 3d Motif Atlas. *RNA*, 19(10):1327–1340, January 2013.

[3] Vladimir Reinharz, Antoine Soulé, Eric Westhof, Jérôme Waldispühl, and Alain Denise. Mining for recurrent long-range interactions in rna structures reveals embedded hierarchies in network families. *Nucleic Acids Research*, 46(8):3841–3851, 2018.

[4] Robert M. Dirks and Niles A. Pierce. An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots. *Journal of Computational Chemistry*, 25(10):1295–1304, 2004.

[5] Craig L. Zirbel, James Roll, Blake A. Sweeney, Anton I. Petrov, Meg Pirrung, and Neocles B. Leontis. Identifying novel sequence variants of RNA 3d motifs. *Nucleic Acids Research*, 43(15):7504–7520, September 2015.

[6] José Almeida Cruz and Eric Westhof. Sequence-based identification of 3d structural modules in rna with rmdetect. *Nature methods*, 8(6):513, 2011.

[7] Ronny Lorenz, Stephan H. Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, 6:26, November 2011.

[8] Kengo Sato, Yuki Kato, Michiaki Hamada, Tatsuya Akutsu, and Kiyoshi Asai. IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. *Bioinformatics*, 27(13):i85–i93, July 2011.

[9] Vladimir Reinharz, François Major, and Jérôme Waldispühl. Towards 3d structure prediction of large RNA molecules: an integer programming framework to insert local 3d motifs in RNA secondary structure. *Bioinformatics*, 28(12):i207–i214, June 2012.

[10] Audrey Legendre, Eric Angel, and Fariza Tahi. Bi-objective integer programming for RNA secondary structure prediction with pseudoknots. *BMC Bioinformatics*, 19:13, January 2018.

[11] IBM ILOG. CPLEX: CPLEX Optimizer (academic license). `https://www.ibm.com/analytics/optimization-modeling-interfaces`, 2018.

[12] Mirela Andronescu, Vera Bereg, Holger H Hoos, and Anne Condon. Rna strand: the rna secondary structure and statistical analysis database. *BMC bioinformatics*, 9(1):340, 2008.