

Aligning Covariance Models

Lu Gram, Rune Lyngsø, Jotun Hein

October 11, 2009

1 Introduction

1.1 RNA secondary structure

While RNA primary structure describes solely the string of bases, RNA secondary structure includes local intra-molecular interactions between pairs of bases, which form weak hydrogen bonds. Two such bonded bases are called *paired*. Thus RNA secondary structure can be represented as a sequence of bases annotated with lines between paired bases. A more intuitive representation displays RNA secondary structure in a two-dimensional format with paired bases forming *stems* and unpaired bases forming *loops*. Finally secondary structures without crossing pairs can be represented in a tree format. All three representations are shown in Figure 1.

1.2 Covariance models

Covariance models are generalisations of Profile Hidden Markov Models (HMM). Where Profile HMMs summarise a family of sequences using a HMM, covariance models summarise a family of sequences using a stochastic context-free grammar (SCFG). This allows for the treatment of the long-distance dependencies required for recognition and alignment of RNA secondary structures. However, due to the limited computational power of SCFGs, only nested dependencies are allowed.

Now, Profile HMMs are sequences of blocks. Each block, apart from the first and the last, contains *insert*, *delete* and *match* states which each contain different transition and emission probabilities. The first block contains a start state and the last block an end state. Since Profile HMMs are used to model DNA sequences at sequence level, we essentially only need one type of block corresponding to the familial distribution of nucleic acids at a particular location in the DNA sequence.

Further, we only need insert, delete and match states, corresponding to extra nucleic acids, missing nucleic acids and present nucleic acids at a particular position. With RNA secondary structures, we classify positions into left, right and pair positions resp. corresponding to the in-going (closest to the 5' end), out-going (closest to the 3' end) and paired (both) positions within an RNA secondary structure. Thus we have the following blocks:

- Left and right singlet blocks correspond to single bases in the consensus RNA structure.

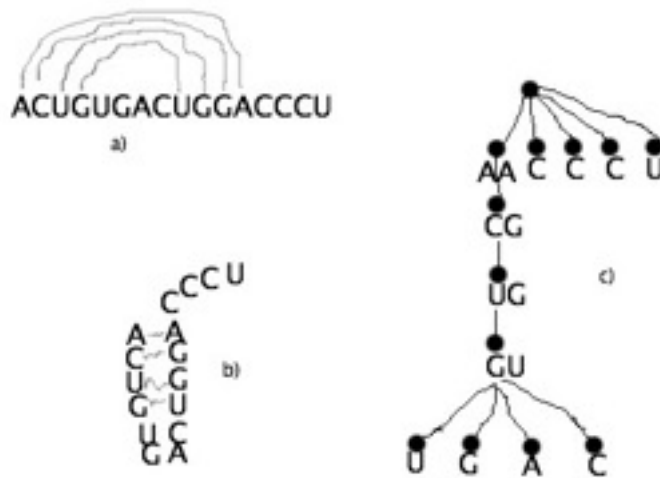


Figure 1: Three representations for RNA secondary structure: a) Linear b) "Geometric" c) Tree-form

- Pair blocks correspond to paired bases in the RNA structure
- Root block does not correspond to any base or base pair, but simply starts the whole model off
- END block similarly does not correspond to any base, but ends a branch of the model
- Bifurcation blocks are used in multiloops to allow multiple pairs linked inside a multiloop. The immediate descendants of a bifurcation block is always a left branch block and a right branch block.

In addition, we have the following states (See Figure 2):

- *Insert left* corresponds to an extra base at a left position
- *Insert right* corresponds to an extra base at a right position
- *Delete* deletes single bases in singlet blocks
- *Match left* matches a base at a left position or deletes the right base in a pair
- *Match right* matches a base at a right position or deletes the left base in a pair
- *Match pair* matches a pair position

Finally, there are uniqueness constraints on the order of appearance of blocks - since left singlet blocks before and after right singlet blocks represent the same consensus structure, they are restricted to occur only before right singlet blocks. Further, right singlet blocks immediately below branch blocks correspond to left

singlets in the neighbouring branch, so we restrict the singlets below branch blocks to be of the left type only. Further, left singlets below left branch blocks correspond to left singlets above the bifurcation node, to no singlets are allowed immediately below a left branch block. Further, singlet nodes before an END block must be left singlet nodes since they correspond to a loop at the end of a stem. Finally, each branch block must contain at least one pair block - otherwise, the branch block could be removed entirely since the branch simply represents a linear sequence of bases.

While evolutionary processes on RNA structures themselves have been proposed[4], this project has the goal of creating and simulating an evolutionary process on families of RNA structures by evolving the corresponding covariance model. This leads in turn to an alignment algorithm for covariance models.

2 Aim

To design and implement an alignment algorithm for aligning covariance models through an evolutionary process. To test the algorithm on covariance models from the Rfam database[2].

3 Designing an evolutionary process

3.1 Outline

Due to the complexity of defining evolutionary processes on covariance models, we restricted ourselves to two operations only: The insertion/deletion of a single base and the pairing/unpairing of a single base pair. Given insert/delete rates of ϵ and μ resp. and pair/unpair rates of ν and λ resp., the equilibrium distribution for the joint number of bases in total m and the number of pairs n is $(\epsilon/\mu)^m(1 - \epsilon/\mu)(\nu/\lambda)^n(1 - \nu/\lambda)$ (this is clear, when we considered detailed balance on individual RNA structures). However, while insertions and deletions result in fairly straightforward operations on covariance models, namely the insertion or deletion of the relevant singlet block, pairing and unpairing operations result in much less transparent results (see Section 3.2). Once we have defined the random process for creating structural changes in the consensus RNA structure, we need to define a random process for altering the parameter values themselves. This process needs to preserve $\sum_i p_i = 1$ for all outgoing transition and emission probabilities. An easy way to define a random process on the probability distribution (p_i) is to let $x_i^2 = p_i$. Then evolving (x_i) as Brownian motion on a hypersphere automatically yields $1 = \sum_i x_i^2 = \sum_i p_i$. Finally, once we have a path linking two covariance models, we want to sample further paths between them using Markov Chain Monte Carlo in such a way that we obtain a representative sample from the space of paths. This will allow us to perform inference, estimate our posterior certainty of homology and identify more and less definitely homologous regions.

3.2 Operations on Covariance Models

The covariance models stored in the Rfam database all use bifurcation nodes. However, for our purposes of defining an evolutionary process, it is more useful to

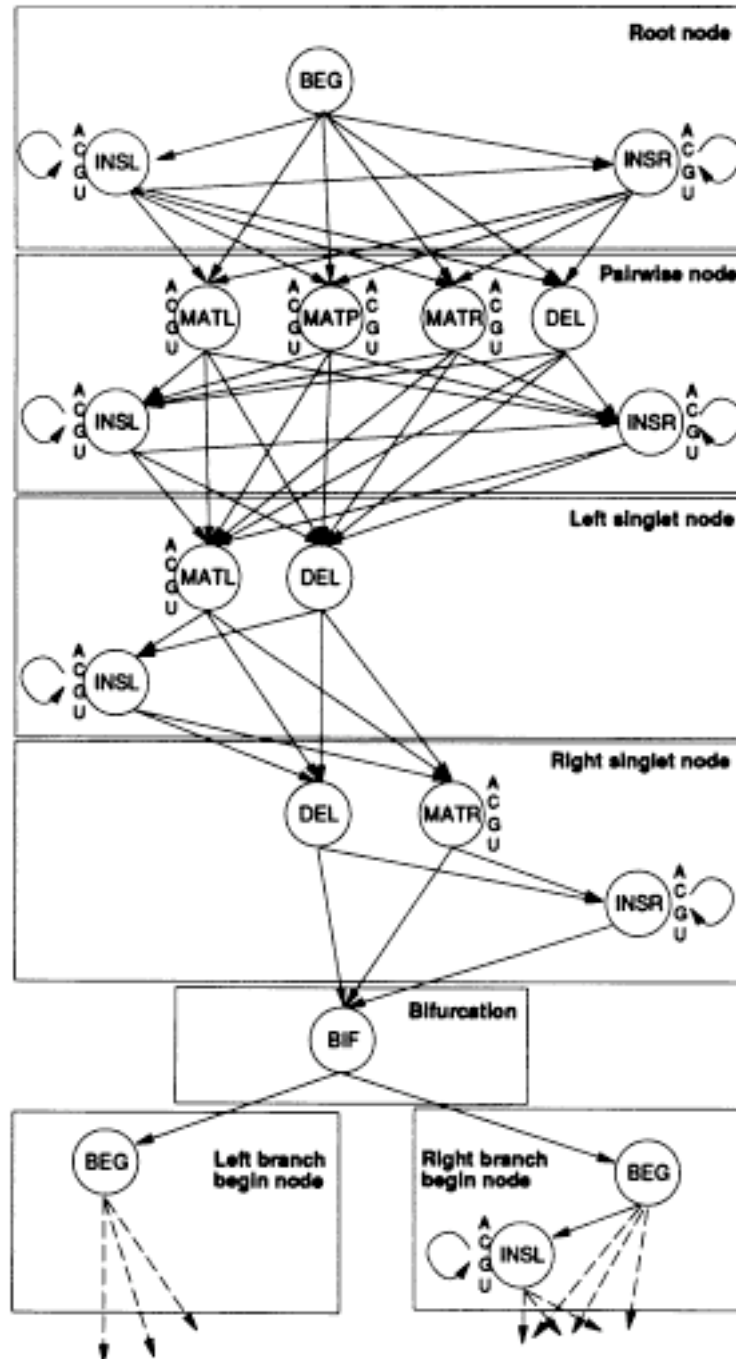


Figure 2: Layout of covariance model

think in terms of multifurcation nodes instead. So we generalise a multifurcation node to be a node with a single left branch and a positive number of right branches all initiated with left branch and right branch blocks. Right after loading a covariance model from the Rfam database, we therefore convert them to a "canonical form" where all consecutive bifurcation nodes are merged to multifurcation nodes.

First we need to define what happens to a covariance model when we insert/delete bases and pair/unpair base pairs. Inserting and deleting bases is easy. When we insert a base in the RNA structure, we insert a left singlet block in loops or on the outgoing side of stem regions i.e. the side of the of the stem closest to the 5' end. When we insert a base on the side of the stem region closest to the 3' end, we insert a right singlet block. When we remove a base, we remove the corresponding singlet block. However, there are important issues that need to be addressed about the change in transition probabilities when creating or removing blocks, since it is not clear how to create new transition probabilities in a meaningful way, especially when the number of probabilities needed can change radically (e.g. when we insert or delete a singlet block in between two pair blocks).

When we pair and unpair bases, it's more complicated. When we pair bases, we must take several cases into account. Firstly, we can either:

- Form pairs between bases corresponding to singlet blocks above a multifurcation block
- Form pairs between one base corresponding to a singlet block above a multifurcation block and singlet block in one of the branches
- Form pairs between two singlet blocks in different branches
- Form pairs between bases above a pair block
- Form pairs between bases before an END block

In the first case, we must then further subdivide into three cases. We assume throughout that x is an ancestor of y in the covariance model:

- Form pair between left singlet x and right singlet y . In this case, we create a new pair block, whose immediate descendants will consist of all the left singlets below x , all the right singlets *below* y . Any right singlets between x and y will have to be moved up *above* the new pair.
- Form pair between left singlet x and left singlet y . Create a new pair block, whose descendants will be all left singlets between x and y followed by an END block. Create a new multifurcation block, move all left singlets before x and all right singlets after y up before the new multifurcation block and let the new pair block start a left branch.
- Form pair between right singlet x and right singlet y . Create a new pair block, whose descendants will be all the right singlets between x and y . However, now take these right singlets, invert their order of appearance and convert them to left singlets. Append an END block. All right singlets below y will also have to be inverted in order and converted to left singlets, then put before the pair. This becomes a new right branch of a multifurcation.

In the second and third cases, we either end up adding a pair to one of the branches or replacing a subset of branches with a single branch containing a new multifurcation with attendant reshuffling of singlet nodes. This multifurcation contains all the "enclosed" branches. In the fourth case, we simply add another pair if we pair a left and right singlet and create a multifurcation if we pair two left singlets or two right singlets. In the last case, we simply add another pair. For further information see the Java program CovModel which deals with all the special cases. It contains classes and functions for creating and manipulating covariance models by inserting/deleting bases and unpairing/pairing base pairs.

4 Outstanding issues

The following is only a modest first step towards creating an alignment algorithm on covariance models. Several unsolved challenges remain to be addressed:

- How should transition probabilities be modified when inserting or deleting or moving around blocks? When a block is inserted in between two other blocks, the old transition probabilities must be removed and new transition probabilities added. It is theoretically not always possible to do this in such a way that inserting and immediately deleting a block corresponds to the identity operation.
- The operation of inverting a sequence of right singlets and then converting them all to left singlets does not correspond to any underlying change in consensus RNA structure. Rather it is performed to preserve the uniqueness of representation of covariance models. However, it is mathematically not possible to perform this operation while keeping all sequence probabilities the same - even HMMs are not invertible.
- How should we obtain a good initial path between two covariance models? The problem of finding the shortest edit distance between two RNA secondary structures in terms of pairing/unpairing operations and insertion/deletion of single bases is SNP-hard even in the absence of pseudoknots [3]. One possibility is to perform an alignment rather than finding shortest edit distance, however even this problem is unclear when we take into account evolution of the transition probabilities - if we delete an arbitrarily long chain of singlet blocks we'll change the transition probabilities into the last singlet block at the end, hence introducing unbounded dependencies that need to be taken into account.
- How should we use our sample of paths to obtain the probability of homology? One possibility is to add together the likelihood of the paths obtained (counting unique paths only once). Since our MCMC chain can generate paths with probability proportional to their likelihood, we obtain the most likely paths first. However, this method is wasteful since repeat paths are thrown away. A more efficient method is yet to be found.

Hopefully, when these problems have been successfully addressed, one can move on to implementing and testing the alignment algorithm on the Rfam database.

References

- [1] Eddy, S. R., Durbin, R., "RNA sequence analysis using covariance models", *Nucleic Acids Res.*, 1994.
- [2] Gardner, P.P., Daub, J., Tate, J.G., Nawrocki, E. P., Kolbe, D. L., Lindgreen, S., Wilkinson, A. C., Finn, R. D., Griffiths-Jones, S., Eddy, S. R., Bateman, A., "Rfam: updates to the RNA families database", *Nucleic Acids Research*, 2008.
- [3] Herrbach, C., Denis, A., Dulucq, S., "Average complexity of the Jiang-Wang-Zhang pairwise tree alignment algorithm and of a RNA secondary structure alignment algorithm", *Elsevier*, 2008.
- [4] Holmes, I., "A probabilistic model for the evolution of RNA structure", *BMC Bioinformatics*, 2004.