CMfinder—A Covariance Model Based RNA Motif Finding Algorithm: Appendix, additional technical details about the algorithm

Zizhen Yao, Zasha Weinberg, Walter L. Ruzzo

December 11, 2005

1 Heuristics to construct the initial alignment

1.1 Candidate selection

We used RNAfold in the Vienna package [1] to compute the minimal free energy for all subsequences in a given sequence. The ones whose length and number of stem-loops are within the range (default 30 $\sim$ 100 bases, 1 $\sim$ 2 stem loops), and are locally optimal (base paired at the ends, with no lower-energy states by extending or shrinking 2 bases at the ends) are selected. They are then sorted by the energy scaled by sequence length, and candidates are selected from the top of the list. We allow overlapped candidates as long as one of them is significantly longer than the other one.

1.2 Selecting conserved candidates and candidate alignment

We first compute a pairwise distance matrix for all candidates from all sequences using the tree-edit algorithm implemented in RNAdist in the Vienna package. We modified the program such that the comparison is at the base/base pair level, and the simple edit distance model used is shown below:

$$\begin{pmatrix}
0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
2 & 0 & 2 & 2 & 2 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
2 & 2 & 0 & 2 & 2 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
2 & 2 & 2 & 0 & 2 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
2 & 2 & 2 & 2 & 0 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
2 & \infty & \infty & \infty & \infty & 0 & 1 & 1 & 1 & 1 & 1 & \infty \\
2 & \infty & \infty & \infty & \infty & 1 & 0 & 1 & 1 & 1 & 1 & \infty \\
2 & \infty & \infty & \infty & \infty & 1 & 1 & 0 & 1 & 1 & 1 & \infty \\
2 & \infty & \infty & \infty & \infty & 1 & 1 & 1 & 0 & 1 & 1 & \infty \\
2 & \infty & \infty & \infty & \infty & 1 & 1 & 1 & 1 & 0 & 1 & \infty \\
2 & \infty & \infty & \infty & \infty & 1 & 1 & 1 & 1 & 1 & 0 & \infty \\
2 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0
\end{pmatrix}
\begin{matrix}
- \\ A \\ C \\ G \\ U \\ GC \\ CG \\ AU \\ UA \\ GU \\ UG \\ other
\end{matrix}$$

The edit distance model above forbids single bases to be aligned with base pairs, therefore, penalizes the alignment heavily for incorrect base pairs. We intend to replace it with more elaborate edit distance model [2]. The edit distance of two RNAs are scaled by the squareroot of their lengths to weight comparisons of RNAs with different sizes properly. Given the distances between all pairs of candidates, we use the following heuristics to select a set of conserved candidates: First, for each sequence i and each of its candiates c_{ij}, we locate its closest match c_{kl} in sequence k, where $l = \arg\min_{l'} dist(c_{ij}, c_{kl'})$. Then for c_{ij}, we compute the sum of distances to its closest matches in all sequences $d_{ij} = \sum_{k \neq i} dist(c_{ij}, c_{kl})$. The candidate with the minimal distance d_{ij} is chosen as the "consensus candidate". Then we iteratively choose one candidate from all the remaining sequences such that its sum of distances to the previously chosen candidates is minimized. If this distance is above some threshold, i.e., the remaining candidates are all significantly different from the chosen ones, the process terminates. Since some sequences may not contain the motif at all, this technique prevents contamination by unlikely candidates. Finally, we construct the initial alignment based on pairwise tree-alignment between each chosen candidate and the consensus candidate.

After the first initial alignment is determined, we select the next "consensus candidate" from the unchosen candidates in the same manner. Other members except the consensus candidates are allowed to

appear in multiple initial alignments in case they are assigned to the wrong alignments initially. However, we forbid significant overlap between two initial alignments (over 40% candidates in common) as they tend to converge to the same solution.

2 The EM algorithm

The following terms are used to formulate the EM algorithm:

- N: the total number of sequences.
- $S = (s_i)_{1 \le i \le N}$: the input sequences.
- m: the number of candidates in each sequence.
- $C_i = (c_{ij})_{1 \le j \le m}$: the candidate set of sequence s_i.
- M: the motif CM.
- B: the Null model (all columns follow i.i.d. multinomial distributions).
- $\Gamma = (M, B, \gamma)$: the finite mixture model, where γ is the mixture probability that a sequence contains a motif.
- $X_i = (x_{ij})$: the occurrence of the motif in C_i ($x_{ij} = 1$ if c_{ij} is a motif, and $x_{ij} = 0$ otherwise).
- $\Pi_i = (\pi_{ij})$: the alignments of candidates C_i with M.
- $D = (L_1, L_2, ..., L_l)$: the sequence alignment. L_i: a column.
- $\sigma = (\alpha, \beta)$: consensus secondary structure for D.
  α: a set of single-stranded columns. β: a set of base paired columns.

The goal is to find $\Gamma = (M, B, \gamma)$ that maximises the log likelihood

$$\log P(S|\Gamma) = \log \prod_i \sum_{X_i} \sum_{\Pi_i} P(s_i, X_i, \Pi_i|\Gamma)$$

The E-step estimates hidden variables X_i and Π_i, and the M-step updates M and γ. Note that our framework differs significantly from the inside-outside algorithm which estimates the CM parameters from unaligned sequences. Although both are EM algorithm using unaligned sequences as input, the inside-outside algorithm assumes that the structure of the model is known and fixed, and performs global alignment. Essentially, it calculates the probability that a state/transition is used by summing over all possible parses involving the given state/transition, and uses such probability estimates to update CM parameters. In our framework, we need to determine where the motifs are located, thus introducing an addition hidden variable X_i; in addition, due to relative poor quality of the initial model, CM structure needs to be refined during iteration. Thus, the following algorithm description will focus on these two aspects.

2.1 E-step

The motif occurrences are estimated using a finite mixture model. Assuming zero or one motif occurrence per sequence, the probabilities that a motif candidate is an instance of a motif can be computed as:

$$\begin{aligned} P(x_{ij} = 1|\Gamma) &= \frac{f_{ij}}{f_{i0} + \sum_{k=1}^{m} f_{i,k}} \\ f_{i0} &= (1-\gamma) P(s_i|X_i = (0, 0, \ldots, 0), \Gamma) \\ f_{ij} &= \lambda P(s_i|x_{ij} = 1, \Gamma) \end{aligned} \quad (1)$$

where $\lambda = \frac{\gamma}{m}$. Let $\widetilde{c_{ij}}$ be the region in sequence i excluding c_{ij}, then

$$P(s_i|x_{ij} = 1, \Gamma) = P(c_{ij}|M) P(\widetilde{c_{ij}}|B)$$

$$P(s_i|X_i = (0, 0, \ldots, 0), \Gamma) = P(c_{ij}|B) P(\widetilde{c_{ij}}|B)$$

Equation 1 can be rewritten as

$$\begin{aligned} P(x_{ij}=1|\Gamma) &= \frac{\lambda P(c_{ij}|M)P(\widetilde{c_{ij}}|B)}{(1-\gamma)P(s_i|X_i=(0,0,\ldots,0),\Gamma)+\sum_{k=1}^{m}\lambda P(c_{ik}|M)P(\widetilde{c_{i,k}}|B)} \\ &= \frac{\lambda P(c_{ij}|M)/P(c_{ij}|B)}{1-\gamma+\sum_{k=1}^{m}\lambda P(c_{i,k}|M)/P(c_{i,k}|B)} \end{aligned} \quad (2)$$

Ideally, we should compute $P(c_{ij}|M)$ by inside algorithm. The inside algorithm computes the probability of a sequence given a covariance model by summing over probabilities of all possible alignment paths. It recursively fills a three-dimensional dynamic programming matrix with values $\alpha_v(i,j)$, where $\alpha_v(i,j)$ is the summed probabilities of all parse subtrees rooted at state v for subsequence $x_i,\cdots,x_j$. However, as we discussed in the Methods section, a specific alignment is needed to infer the covariance model in the M-step. Thus, we use the Viterbi algorithm (also known as the CYK algorithm in the context of covariance models) to compute $P(\pi_{ij}|M)$, the probability for the optimal alignment path given the covariance model, as an approximation to $P(c_{ij}|M)$. Similar to the inside algorithm, the Viterbi algorithm is a dynamic programming algorithm that fills a three-dimensional table, but instead of taking the sum of probabilities for all parse substrees, it computes the maximum probability of all substrees. The resulting probability $P(x_{ij}=1)$ can be interpreted as the probability for c_{ij} to be a motif instance given alignment π_{ij}.

2.2 M-step

The M-step has been largely explained in the Method Section except some details in the dynamic programming algorithm to predict a set of base pairs that maximizes the sum of K_{ij} (K_{ij} is a combination of a covariance term and a partition function term). To avoid prediction of isolated unreliable base pairs, we introduce a threshold parameter for helices; if the K_{ij} terms, summed over base pairs in a given helix, falls below this threshold, the helix is not reported as part of the optimal structure.

3 Combining Motifs

Our method works best with relatively short motifs ($\leq$ 100 bases). Of course, some RNAs can be much longer. In these cases, it is very likely that our top scoring motifs cover different regions of the RNA, so we try to combine separate short motifs into longer ones. To merge two motifs, we simply concatenate them, and the regions between the two motifs are aligned by padding with gaps. If two motifs overlap, the region of overlap is assigned to the motif with greater number of affected base pairs, while the involved base pairs in the other motif are removed. The merged motif is then iteratively refined using the EM module. To determine which two motifs should be combined, and the order of combination, we applied a greedy hierarchical merging heuristic: for every pair of motifs, we compute a rough estimate of the alignment score after the combination, which is the sum of their alignment scores penalized by the number of gaps or overlaps between them. The highest scoring pair of motifs will be combined and removed from future consideration. If the combined motif improves the scores relative to each of the two motifs, then it is added to the set of motifs for further combination, otherwise, it is removed.

References

[1] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structure. *Chemical Monthly*, 125:167–88, 1994.

[2] Robbie J. Klein and Sean R. Eddy. RSEARCH: finding homologs of single structured RNA sequences. *BMC Bioinformatics*, 4(1):44, 2003.