Corresponding author's email: clote@bc.edu

PSB Session: Computational Studies of non-coding RNAs

The submitted paper contains original, unpublished results, and is not currently under consideration elsewhere.

All co-authors concur with the contents of the paper.

RNA structural segmentation

Ivan Dotu
Department of Computer Science, Brown University, Box 1910, Providence, RI 02912.

William A. Lorenz
Department of Biology, Boston College, Chestnut Hill, MA 02467, USA.

Pascal Van Hentenryck
Department of Computer Science, Brown University, Box 1910, Providence, RI 02912.

Peter Clote
Department of Biology, Boston College, Chestnut Hill, MA 02467, USA.
(∗) Corresponding author with email: clote@bc.edu

We describe several dynamic programming segmentation algorithms to segment RNA secondary and tertiary structures into distinct *domains*. Since many non-coding RNA gene finders scan the genome by a moving window method, reporting high-scoring windows, we apply structural segmentation to determine the most likely $5'$ and $3'$ boundaries of precursor microRNAs. When tested on all precursor microRNAs of length at most 100 nt from the Rfam database, benchmarking studies indicate that segmentation determines the $5'$ boundary with discrepancy having mean -0.640 (stdev 15.196) and the $3'$ boundary with discrepancy having mean -0.266 (stdev. 17.415). This yields a sensitivity of 0.911 and positive predictive value of 0.906 for determination of exact boundaries of precursor microRNAs within a window of approximately 900 nt. Additionally, by comparing the manual segmentation of Jaeger et al. with our optimal structural segmentation of 16S and 16S-like rRNA of *E. coli*, rat mitochondria, *Halobacterium volcanii*, and *Chlamydomonas reinhardii* chloroplast into 4 segments, we establish the usefulness of (automated) structural segmentation in decomposing large RNA structures into distinct domains.

Availability: Source code for all algorithms is available at `http://bioinformatics.bc.edu/clotelab/`.

Keywords: non-coding RNA gene finder, segmentation algorithm,secondary structure, tertiary structure, RNA domain

1. Introduction

Several groups, such as Benaola-Galván et al.,[2] Román-Roldán et al.,[13] and Li et al.,[8–10] have developed recursive segmentation algorithms with the goal of segmenting chromosomal regions in order to detect isochores, CpG islands and other broad genomic features. The underlying idea of such divide-and-conquer recursive segmentation algorithms is similar to that of C4.5 decision trees, cf. Quinlan,[12] and depends on repeatedly splitting a segment into left and right halves in order to maximize the Jenson-Shannon divergence

$$JS(L,R) = H(W) - \frac{m}{n}H(L) - \frac{n-m}{n}H(R).$$

Here L, R are the left and right segments of the whole segment W, and the lengths of L, R, W are respectively $m, n-m, n$. Figure 1 depicts the pseudocode corresponding to this approach. We refer the reader to Clote and Backofen[3] for detailed explanation and full pseudocode of this algorithm.

While this method has been applied to the detection of broad features of chromosomal DNA,[2,8–10,13] other segmentation algorithms in the literature have been introduced by Finkelstein and Roytberg[4] (dynamic programming), and Schmidler et al.[14] (Bayesian a posterior method). Applications of the dynamic programming segmentation algorithm of Finkelstein and Roytberg[4] have been given by Sunyaev et al.[15] for multiple alignments of proteins, while applications of the Bayesian a posterior method have been presented by Schmidler et al.[14] to predict protein secondary structure α-helices and β-sheets given the primary sequence information.

In this paper, we describe our re-discovery and extension of the Finkelstein-Roytberg dynamic programming segmentation algorithm, which we apply to segment 3-dimensional RNA structures into *domains*, and use to detect the boundaries of certain non-coding RNA genes within high scoring windows, as determined by many moving-window genome scanning algorithms.

```
void segment( int i, int j, double s) {
   max=0
   for k=i to j-1{
      $L = w_i \cdots w_k$
      $R = w_{k+1} \cdots w_j$
      if statistical significance of splitting $L, R$ exceeds $s$ then
         if $JS(L,R) > \max$ then
            $\max = JS(L,R)$
            $x = k$
      }
   print x
   segment(i,x,s)
   segment(x+1,j,s)
}
```

Fig. 1. Pseudocode for recursive segmentation algorthm of Román-Roldán et al.[13] Note that one of the difficulties of this approach consists in determining a minimum threshold s, below which segmentation is discontinued.

2. Methods

The problem we consider consists in segmenting a sequence $S = \langle s_1, \ldots, s_n \rangle$ into a number of consecutive subsequences (called segments) $S_1, \ldots, S_k$. (The sequence S is thus the concatenation of $S_1, \ldots, S_k$.) Each segment S_i is associated with a fitness value $f(S_i, S)$ which only depends on the elements in S_i and those in $S \setminus S_i$, not on the segmentation itself. For instance, in practical applications, such a function will often be expressed in terms of two functions g and h as follows:

$$f(S_i, S) = \sum_{x,y \in S_i, x \neq y} g(x,y) - \sum_{x \in S_i, y \in S \setminus S_i} h(x,y).$$

Our goal is to find a segmentation $S_1, \ldots, S_k$ that maximizes the sum of the fitness values, i.e.,

$$\sum_{i=1}^{k} f(S_i, S).$$

Observe that the number of segments k is not fixed and is chosen to maximize the overall fitness. In the following, we also use $f_{i,j}$ to denote $f(\langle s_i, \ldots, s_j \rangle, S)$.

2.1. *Dynamic programming using quadratic time and quadratic space*

We now present an $O(n^2)$ algorithm to solve this problem. The key idea underlying the algorithm is to reason about partial segmentations which cover prefixes $(s_1, \ldots, s_k)$ but whose fitness values are computed with respect to the entire sequence S. Obviously, when $k = n$, we obtain a solution to the original problem.

The algorithm is based on a recurrence relation on the starting positions and lengths of the last segment in an optimal (partial) segmentation. More precisely, $F[\ell, s]$ denotes the fitness value of the best partial segmentation whose last segment has length ℓ and starting position x. The base case corresponds to $x = 1$ and is given by

$$F[\ell, 1] = f_{1,\ell}$$

for $1 \leq \ell \leq n$. The recursive case for $1 < x$ is given by the formula

$$F[\ell, x] = f_{x,x+\ell-1} + \max\{F[i,j] : j + i = x\}$$

The left part of the sum is the fitness value of the last segment. The right part is the fitness value of the best partial segmentation that ends at $x - 1$. It is obtained by considering the fitness values of all the partial segmentations of $\langle s_1, ..., s_{x-1}\rangle$. By induction, these fitness values are associated with their last segments, i.e., segments that start at some position j, have some length i, and end at position $x - 1$. The fitness value of the optimal segmentation of S is then given by

$$\max\{F[\ell, x] : \ell + x = n + 1, \ell > 1, x \geq 1\}.$$

Given the entry (ℓ^*, x^*) with maximal fitness value $F[\ell^*, x^*]$, the set of starting positions $st[\ell^*, x^*]$ of the segments in the best segmentation can be traced backwards from using the following recurrence

$$\begin{aligned} st[\ell, x] &= \{x\} \cup st[x - p, p] \\ &\qquad \text{where } p = \max\{p' : F[x - p', p'] = \max\{F[i,j] : j + i = s\}\}; \\ st[\ell, 1] &= \emptyset \end{aligned}$$

which, at each step, retrieves the last segment $\langle s_{x-p}, \ldots, s_{x-1}\rangle$ of the optimal partial segmentation.

We now argue that these recurrence relations can be computed by an $O(n^2)$ dynamic programming algorithm. First observe that the expression

$$\max\{F[i,j] : j + i = x\}$$

must only consider $x-1$ segments since $j \geq 1$ and $i \geq 1$, i.e., there are only $O(x)$ pairs to consider. Moreover, observe that this expression does not depend on ℓ in the recurrence relation and hence can be computed once for all entries $F[1, x], \ldots, F[n - x, x]$. As a result, the dynamic programming algorithm runs in $O(n^2)$ provided that the expression is computed once at the beginning of each column. Note also that the index p in the recurrence for st can be computed during the forward computation, so that the backward computation takes only $O(n)$ time.

Note that this algorithm can yield the maximum, minimum, and average fitness of all segments; however, the space required is quadratic. In the next section, we describe a linear space algorithm.

2.2. *Dynamic programming using quadratic time and linear space*

Given the complete segment $S = s_1, \ldots, s_n$, let $F(i)$ designate the maximum fitness over all segmentations of $s_1, \ldots, s_i$. Straightforwardly,

$$F(i) = \begin{cases} 0 & \text{if } i = 0 \\ \max(f(1,i), \max_{1 \leq k < i} F(k) + f(k+1, i)) & \text{else.} \end{cases} \tag{1}$$

It can be seen how the maximum fitness of S is given by $F(n)$, and by means of tracebacks, we obtain the obtimal segmentation. Computation time is obviously quadratic in n, while space is linear in n. This latter version of the segmentation algorithm turned out to be equivalent to that of Finkelstein and Roytberg,[4] displayed in equation (1).

2.3. *Parametric dynamic programming method*

In this section, we describe a new algorithm that computes, given an RNA sequence (structure) and integer K, the optimal segmentation into k segments, for each $1 \leq k \leq K$. This algorithm runs in time $O(n^2 k)$ and space nk.

The underlying idea of the algorithm described in this section is to maintain separately indexed tables $F[m, i]$ for the optimal fitness over all segmentations of $[1, m]$ into i segments. Letting $f(i, j)$ denote the base fitness of segment $[i, j]$, as defined, for instance, by

$$f(i,j) = \sum_{i \leq x < y \leq j} wt_1 \cdot p_{x,y} - sum_{x \in [i,j]} \sum_{y \notin [i,j]} wt_2 \cdot p_{x,y}$$

```
int[n][n] parametricSegmentation(rna,f,numSegments){
  /*------------------------------
   rna is RNA sequence, f is base fitness function.
  ---------------------------------------------------*/
  n = len(rna); SplitPoints = $\emptyset$
  for $d = LOWER$ to $n$
    for $num = 2$ to $\min(numSegments, d/LOWER)$
      for $m = (num-1) \cdot LOWER + 1$ to $d - LOWER - 1$
        val = F[m,num-1]+f[m+1,d]
        if $val > \max$
          $splitPoint = m$
          $\max = val$
      F[d,num]  = max
      SplitPoints[d,num] = splitPoint
  return SplitPoints  //Using SplitPoints array, one can perform traceback
  }
```

Fig. 2. Pseudocode for parametric segmentation algorithm to compute optimal fitness $F[d,k]$ over all segmentations of $[1,d]$ into k segments. Note how bounds for minimum segment size (LOWER) and maximum segment size (UPPER) can easily be accommodated within such segmentation algorithms.

we inductively define $F[m,i] = \max_{1 \leq k < m} F[k,i-1] + f(k+1,m)$. (See Figure 2 for pseudocode of algorithm.) Clearly the run time of parametric segmentation is $O(n^2 \cdot K)$ and the space requirement is $O(n \cdot K)$, when computing optimal segmentations of $[1,n]$ into k segments, for all $k \leq K$.

2.4. *Optimal fitness of all segmentations of subwords*

In this section, we describe a cubic time algorithm to compute the optimal segmentation, simultaneously for all subwords $[i,j]$, where $1 \leq i \leq j \leq n$. This algorithm is inspired by the Nussinov-Jacobson algorithm,[11] which determines the secondary structure having a maximum number of base pairs. (See Figure 3 for the pseudocode of this algorithm.) By using this algorithm, where the base fitness function f is defined from the contact map obtained by RNAview,[16] one could produce segmentations where low scoring initial portions $[1,i-1]$ and low scoring terminal portions $[j+1,n]$ are dropped, thus leaving a segmentation of subword $[i,j]$. The manual segmentations of Jaeger et al.[5] described in the Results section appears to be of this type.

2.5. *Fitness Functions*

We have considered different fitness functions for RNA secondary structure, all of them fitting in the following scheme:

$$F(i,j) = \sum_{i \leq x < y \leq j} w_1 \cdot p_{x,y} - \sum_{x \in [i,j]} \sum_{y \notin [i,j]} w_2 \cdot p_{x,y}.$$

where $F[i,j]$ is the fitness function of segment $[i,j]$ and $p_{x,y}$ can be the following:

- The base pair probability between nucleotides x and y as computed by RNAfold -p.
- The existence (or not) of a base pair between nucleotides x and y as computed by RNAview.

The pseudocode to compute the fitness function for base pairing probabilities (and equivalently for contact maps) is depicted in Figure 4.

We have also considered a 3D fitness function (which can also be used for proteins or other molecules) which consists on minimizing the normalized volume (by computing a tesselation with Qhull[1]). The fitness function of segment $[i,j]$ is thus the normalized volume as calculated by Qhull.

```
int[n][n] segmentation(rna){
  // rna is RNA sequence, f is pre-computed base fitness function.
  n = len(rna); SplitPoints $= \emptyset$
  for $d = 1$ to $n - 1$
    for $i = 1$ to $n$
      $j = i + d$
      if $(j > n)$ then break
      $\max = f[i, j]$
      for $k = i$ to $j - 1$
        $val = F[i, k] + f[k + 1, j]$
        if $val > \max$ then
          $\max = val$
          $splitPoint = k$
      F[i,j]     = max
      SplitPoints[i,j] = splitPoint
  return SplitPoints
  }
```

Fig. 3. Algorithm to determine optimal segmentation of each subsequence $[i, j]$, with run time $O(n^3)$ and space $O(n^2)$. This algorithm is inspired by the Nussinov-Jacobson algorithm,[11] which determines the secondary structure having maximum number of base pairs. Assuming the base fitness function f has been precomputed, this algorithm the fitness $F[i, j]$ for the optimal segmentation of each subsequence $[i, j]$. The optimal segmentation can be computed by traceback using the information from *SplitPoints*.

```
void fitness(rna){
  using RNAfold -p determine base pairing probabilities $p_{i,j}$
  n = len(rna)
  for $d = 1$ to $n$
    for $i = 1$ to $n$
      $j = i + d$
      if j>n then break
      if i==j
        sum = 0.0
      else //$i < j$
        sum = f[i,j-1]
      for $k = 1$ to $n$
        if $i \leq k < j$
          sum += $(w_1 + w_2)\cdot$ P[k][j]
        else if $k < i$
          sum -= $w_2 \cdot p_{k,j}$
        else if $k > j$
          sum -= $w_2 \cdot p_{j,k}$
      f[i,j] = sum
  return f
}
```

Fig. 4. The base fitness $f[i, j]$ of segment $[i, j]$ is defined by $\sum_{i \leq x < y \leq j} w_1 \cdot p_{x,y} - \sum_{x \in [i,j]} \sum_{y \notin [i,j]} w_2 \cdot p_{x,y}$, where base pairing probabilities $p_{x,y}$ are computed by RNAfold -p. Straightforward implementation of the formula for $f[i, j]$ requires $O(n^4)$ time. In contrast, this figure depicts pseudocode to compute base fitness function f in time $O(n^3)$.

3. Results

3.1. *Finding Precursor microRNAs*

As previously mentioned, we applied our segmentation to help determine non-coding RNA genes within a window of flanking nucleotides. Many non-coding RNA gene finders use a moving window strategy, where the likelihood that the fixed-size window contents contain a non-coding RNA gene is represented by a numerical

Table 1. Boundary prediction: precursor microRNA from Rfam of size ≤ 100 nt.

	Left Border		**Right Border**		**Stats**	
Parameters	**Mean**	**St Dev**	**Mean**	**St Dev**	**Sensitivity**	**PPV**
$w_1 = 1, w_2 = 0 - 50$	9.984	16.193	-9.486	17.115	0.774	0.990
$w_1 = 1, w_2 = 0 - 100$	10.032	15.814	-10.035	17.441	0.770	0.992
$w_1 = 1, w_2 = 0 - 200$	9.691	15.059	-10.887	17.450	0.765	0.993
$w_1 = 1, w_2 = 0 - 400$	10.206	15.899	-11.038	18.063	0.761	0.992
$w_1 = 0, w_2 = 1 - 50$	-2.453	14.132	1.891	13.149	0.927	0.888
$w_1 = 0, w_2 = 1 - 100$	-1.807	7.102	1.379	11.489	0.969	0.936
$w_1 = 0, w_2 = 1 - 200$	-3.331	11.541	4.199	10.462	0.963	0.887
$w_1 = 0, w_2 = 1 - 400$	-4.113	11.803	3.351	12.122	0.949	0.876
$w_1 = 1, w_2 = 1 - 50$	-0.598	15.612	1.235	14.142	0.922	0.903
$w_1 = 1, w_2 = 1 - 100$	-0.701	9.917	1.428	10.856	0.956	0.935
$w_1 = 1, w_2 = 1 - 200$	-1.492	11.211	1.624	10.344	0.945	0.916
$w_1 = 1, w_2 = 1 - 400$	-1.322	12.006	1.483	12.571	0.935	0.909
$w_1 = 2, w_2 = 1 - 50$	-0.524	15.994	0.125	15.654	0.913	0.905
$w_1 = 2, w_2 = 1 - 100$	0.376	12.380	1.096	13.667	0.934	0.927
$w_1 = 2, w_2 = 1 - 200$	-0.299	15.326	-0.132	14.447	0.920	0.918
$w_1 = 2, w_2 = 1 - 400$	-0.640	15.196	-0.266	17.415	0.911	0.906
$w_1 = 1, w_2 = 2 - 50$	-1.958	12.122	0.846	11.772	0.933	0.908
$w_1 = 1, w_2 = 2 - 100$	-0.846	9.419	1.547	8.970	0.964	0.939
$w_1 = 1, w_2 = 2 - 200$	-2.251	10.180	2.080	9.906	0.953	0.911
$w_1 = 1, w_2 = 2 - 400$	-2.955	11.547	2.168	11.022	0.944	0.894
$w_1 = 5, w_2 = 1 - 50$	0.740	15.887	-0.723	17.444	0.901	0.913
$w_1 = 5, w_2 = 1 - 100$	1.968	15.787	-1.572	17.908	0.886	0.921
$w_1 = 5, w_2 = 1 - 200$	2.524	16.453	-1.482	16.025	0.886	0.927
$w_1 = 5, w_2 = 1 - 400$	2.392	17.011	-2.727	18.053	0.868	0.920
$w_1 = 1, w_2 = 5 - 50$	-2.408	13.061	1.129	12.430	0.931	0.899
$w_1 = 1, w_2 = 5 - 100$	-1.431	8.053	1.203	10.187	0.966	0.939
$w_1 = 1, w_2 = 5 - 200$	-2.997	10.655	3.569	10.482	0.960	0.894
$w_1 = 1, w_2 = 5 - 400$	-3.843	11.888	3.297	11.626	0.949	0.878

score. To that end, we tested our segmentation algorithm to detect precursor microRNA within a window of flanking nucleotides, where the flanking nucleotides were extracted from the EMBL genomic file. Our experiment can be summarized as follows.

- Download all the accession codes for precursor micro RNA, riboswitches and SECIS (only results for precurso microRNA are reported here).
- Download the EMBL data for each of the above with 500 flanking nucleotides on each side (when possible). In some cases, there were fewer than 500 nucleotides to the left, or less than 500 nucleotides to the right, in which case the sequence was skipped.
- Run segmentation algorithm combining the varying the following parameters:
  - flanking nts (50, 100, 200, 400)
  - max segment size(100, 1000 which translate to not having a maximum size in practice)
  - weight combinations w1 w2 (1 0, 0 1, 1 1, 2 1, 1 2, 5 1, 1 5)
  - base pairing probabilities, obtained by RNAfold -p
- Report histograms and measures of accuracy.
- Run segmentation with flanking nucleotides replaced by random combination (permutation)

Tables 1 and 3 show, respectively, the results of our segmentation with and without maximum segment size limit. The main conclusions that can be drawn are the following:

- Certain weight combinations yield very poor results, specially in the case of $w1 = 0$, $w2 = 1$ and $w1 = 1$, $w2 = 0$ which means that both characteristics of inside and cross-segments are necessary.
- Giving a higher weight to cross-segment characteristics does not yield the best results which indicates that the local structure of the precursor micro RNA is stronger than its lack of potentially base pair

Table 3. Boundary prediction: precursor microRNA from Rfam, no size limit.

	Left Border		Right Border		Stats	
Parameters	**Mean**	**St Dev**	**Mean**	**St Dev**	**Sensitivity**	**PPV**
$w_1 = 1, w_2 = 0 - 50$	9.113	16.578	-8.624	17.574	0.782	0.984
$w_1 = 1, w_2 = 0 - 100$	9.325	16.005	-9.341	17.723	0.777	0.989
$w_1 = 1, w_2 = 0 - 200$	9.016	15.235	-10.222	17.661	0.772	0.990
$w_1 = 1, w_2 = 0 - 400$	9.479	16.141	-10.322	18.314	0.769	0.988
$w_1 = 0, w_2 = 1 - 50$	-48.997	0.057	49.994	0.113	1.000	0.467
$w_1 = 0, w_2 = 1 - 100$	-98.990	0.098	100.000	0.000	1.000	0.304
$w_1 = 0, w_2 = 1 - 200$	-199.000	0.000	200.000	0.000	1.000	0.179
$w_1 = 0, w_2 = 1 - 400$	-399.000	0.000	400.000	0.000	1.000	0.099
$w_1 = 1, w_2 = 1 - 50$	-37.814	17.105	39.601	17.305	0.993	0.552
$w_1 = 1, w_2 = 1 - 100$	-58.402	42.203	62.003	42.980	0.978	0.507
$w_1 = 1, w_2 = 1 - 200$	-58.929	72.924	57.685	72.334	0.966	0.604
$w_1 = 1, w_2 = 1 - 400$	-60.941	99.448	65.014	99.134	0.957	0.614
$w_1 = 2, w_2 = 1 - 50$	-28.331	21.168	28.611	22.686	0.980	0.649
$w_1 = 2, w_2 = 1 - 100$	-34.624	39.918	33.177	40.922	0.956	0.666
$w_1 = 2, w_2 = 1 - 200$	-22.457	34.832	25.801	39.003	0.960	0.725
$w_1 = 2, w_2 = 1 - 400$	-25.801	43.108	30.119	45.335	0.948	0.702
$w_1 = 1, w_2 = 2 - 50$	-42.132	13.798	43.605	13.462	0.997	0.514
$w_1 = 1, w_2 = 2 - 100$	-76.897	34.990	80.740	33.833	0.990	0.395
$w_1 = 1, w_2 = 2 - 200$	-123.775	82.962	120.785	84.169	0.989	0.385
$w_1 = 1, w_2 = 2 - 400$	-214.867	169.619	211.993	169.320	0.979	0.343
$w_1 = 5, w_2 = 1 - 50$	-11.170	21.748	10.871	21.846	0.943	0.802
$w_1 = 5, w_2 = 1 - 100$	-9.219	26.431	9.563	26.815	0.913	0.821
$w_1 = 5, w_2 = 1 - 200$	-7.762	22.604	7.460	22.576	0.923	0.836
$w_1 = 5, w_2 = 1 - 400$	-6.720	25.015	5.871	24.055	0.905	0.840
$w_1 = 1, w_2 = 5 - 50$	-45.601	10.258	46.775	9.366	0.998	0.488
$w_1 = 1, w_2 = 5 - 100$	-87.682	25.800	91.077	23.983	0.998	0.341
$w_1 = 1, w_2 = 5 - 200$	-174.678	53.153	167.801	62.362	0.999	0.226
$w_1 = 1, w_2 = 5 - 400$	-339.252	118.600	352.521	104.862	0.995	0.136

with other regions in other suboptimal configurations.

- Overall, the weight combination $w1 = 2$, $w2 = 1$ achieves the best results.
- The algorithm is robust to the size of the flanking nucleotides.
- Limiting the maximum size of the segment does impact efficiency. Interestingly, the weight combination $w1 = 5$, $w2 = 1$ performs better in this case. This seems to indicate that a higher weight to inside base pairings is necessary for larger instances since it reinforces its locality, i.e., if there are more nucleotides there are potentially more possibilities of cross-segment base pairings which (in this case), for nucleotides farther away in the primary sequence might not be very significant.

A very useful tool to visualize the quality of the results is to plot the distributions of both left and right end segments of the calculated precursor micro RNA. This information is depicted in Figure 5. Note that both distributions are very similar and they clearly show a higher concentration of segmentations in which the distance from the actual end segment and the calculated one are very close to 0.

It is conjectured that precursor micro RNAs have a very strong local structure with which the flanking nucleotides cannot compete. To prove that our algorithm is sensitive to that local structure (which is consistent with the fact that a higher weight for inside segment yields better results) we have carried out a set of experiments in which we permuted the flanking nucleotides before performing the segmentation. Results of this are shown in Table 5 (where we compare them against the *normal* seuquence, i. e., that with the actual flanking nucelotides), and the distributions are depicted in Figure 6. These results are for weight combination $w1 = 2$, $w2 = 1$ with 400 flanking nucleotides and with no maximum segment size limit. As it can be seen, results are very similar to those for the actual sequence which proves the robustness of our approach.

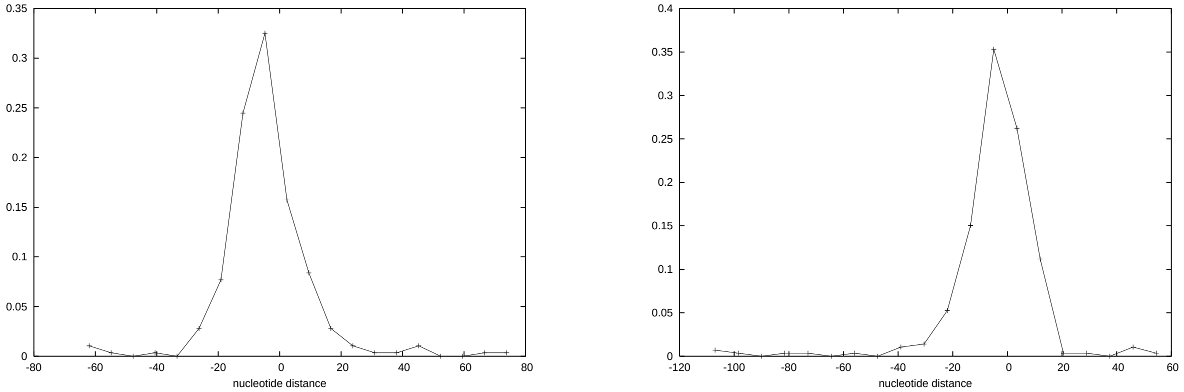


Fig. 5. Histogram of boundary discrepancy for 5′ end (left panel) and 3′ end (right panel) of precursor microRNAs within window having 400 nt flanking both on left and right of microRNA. Analysis performed over all precursor microRNAs from Rfam 9.1 (January 2009, 454 subfamilies).

Table 5. Boundary prediction: permuted versus unpermuted tails of precursor miRNA

	Left Border		**Right Border**		**Stats**	
Parameters	**Mean**	**St Dev**	**Mean**	**St Dev**	**Sensitivity**	**PPV**
$Normal - 50$	-0.524	15.994	0.125	15.654	0.913	0.905
$Normal - 100$	0.376	12.380	1.096	13.667	0.934	0.927
$Normal - 200$	-0.299	15.326	-0.132	14.447	0.920	0.918
$Normal - 400$	-0.640	15.196	-0.266	17.415	0.920	0.918
$Permuted - 50$	1.129	14.073	-2.334	15.948	0.899	0.939
$Permuted - 100$	0.251	11.629	-1.074	14.183	0.928	0.944
$Permuted - 200$	1.180	14.354	0.113	12.329	0.918	0.933
$Permuted - 400$	0.287	14.406	-0.955	15.669	0.910	0.924

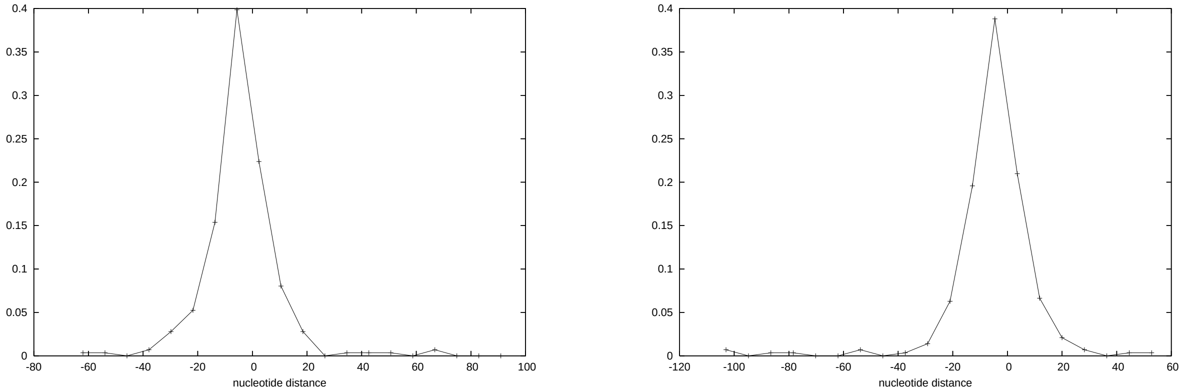


Fig. 6. Histogram of boundary discrepancy for 5′ end (left panel) and 3′ end (right panel) of precursor microRNAs within window having 400 nt flanking both on left and right of microRNA. Analysis performed over all precursor microRNAs from Rfam 9.1 (January 2009, 454 subfamilies) with permuted flanking nucleotides.

3.2. *Finding RNA domains*

Our initial motivation for developing a segmentation algorithm was to determine an automated method to decompose large X-ray structures of RNA, such as PDB code 1FFK, into coherent units, or domains. Also, to segment RNA sequences in which secondary structure is available.

With the intent of benchmarking the accuracy of MFOLD, Jaeger et al.[5] performed a manual segmentation of *E. coli* 16S rRNA, as well as the 16S-like rRNA domains of rat mitochondria, *Halobacterium volcanii*, and *Chlamydomonas reinhardii* chloroplast into 4 segments.

Table 7. Manual and computed segmentations of 16S rRNA.

Organism & method	seg 1	seg 2	seg 3	seg 4	fit 1	fit 2	fit 3	fit 4
E. coli (manual)	27 − 509	515 − 857	866 − 1326	1329 − 1476	0.628	0.623	0.462	0.658
E. coli (computed)	1 − 338	339 − 350	351 − 1132	1133 − 1542	0.399	0.635	0.573	0.570
rat mitochondrial (manual)	20 − 279	279 − 509	526 − 829	829 − 953	0.550	0.459	0.559	0.323
rat mitochondrial (computed)	1 − 459	460 − 484	485 − 928	929 − 953	0.551	0.760	0.785	0.6221
H. volcanii (manual)	21 − 495	501 − 857	865 − 1342	1342 − 1474	0.600	0.618	0.597	0.617
H. volcanii (computed)	1 − 84	85 − 405	406 − 433	434 − 1476	0.551	0.760	0.785	0.622
C. reinhardii (manual)	27 − 509	515 − 857	866 − 1326	1329 − 1476	0.632	0.622	0.596	0.647
C. reinhardii chloroplast (computed)	1 − 754	755 − 1350	1351 − 1413	1414 − 1476	0.480	0.466	0.673	.563

Table 8. Average, min, max fitness over all segments; manual segmentation fitness of 16S rRNA.

Organism & method	avg	min	max	fit 1	fit 2	fit 3	fit 4
E. coli	0.308	−1.000	0.857	0.628	0.623	0.462	0.658
rat mitochondrial	0.190	−1.000	0.857	0.550	0.459	0.559	0.323
H. volcanii	0.292	−1.000	0.857	0.600	0.618	0.597	0.617
C. reinhardii	0.299	−1.000	0.916	0.632	0.622	0.596	0.647

In Table 7 we present results from the manual and optimal segmentation of 16S rRNA into four segments. Optimal segmentation is calculated using base pairing probabilities with weights $w_1 = 2$, $w_2 = 1$ (these weights were determined by previous benchmarking experiments). In that table, column headings, *seg* abbreviates segment, while *fit* abbreviates fitness. The manual segmentation was created by Jaeger et al.,[5] while the computed segmentation used the parametric algorithm described in Figure 2. Note that we could have modified (but did not) the parametric segmentation to discard with no penalty a small initial and final segment. Since this was not done, all computed segmentations begin at nucleotide 1 and end at the last nucleotide, unlike that from the manual segmentation. This explains how a manual segmentation can paradoxically have higher fitness than the computed *optimal* segmentation.

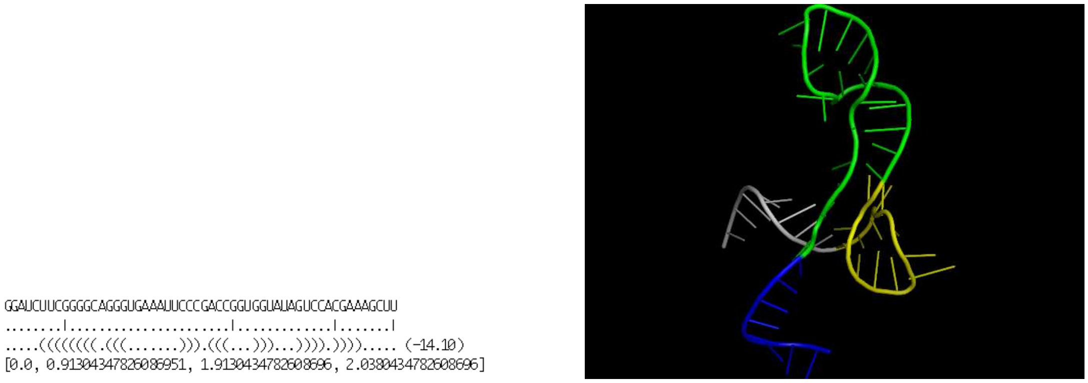
Even though our optimal segmentation does not always resemble the manual segmentation, from Table 8 (which shows average, minimum and maximum fitness for all segments) it can be seen how all manually calculated segments have fitnesses higher than the average. This seems to indicate that our fitness function correlates with reality but that possibly more specific information needs to be added to boost efficiency.

As another example of the application of our segmentation algorithm to divide an RNA sequence into domains we have segmented PDB file 3F4H:X using the fitness function corresponding to contact maps from RNAview.

Figure 7 depicts the results both in text format and as a Pymol image in which different segments appear in different colors. This latter image shows more clearly the division in domains and it appears to be reasonable in light of its 3D representation. Note that segments determined by structural segmentation are not simply α-helices or β-strands.

4. Conclusions

In this paper, we present a rediscovery and extension of dynamic programming algorithm for optimal segmentation. Optimal parametric segmentation appears to be new, as is the computation of the partition function and stochastic sampling (algorithm to be given in final version of paper). Applications of segmentation in the context of RNA include *(i)* an automated method to decompose large RNA 3-dimensional structures into domains suitable for estimating knowledge-based potentials or instead for benchmarking secondary structure algorithms, as done manually by Jaeger et al.,[5] *(ii)* a method to determine the possible $5'$ and $3'$ boundary of non-coding RNA gene found within a window of a genome scanning algorithm. As future work we would like to add other metrics to our fitness function as well as to perform exhaustive benchmarking on 3D segmentation using Qhull. Preliminary results on trans-membrane proteins (Figure 8) show the potential of this



```
GGAUCUUCGGGGCAGGGUGAAAUUCCCGACCGGUGGUAUAGUCCACGAAAGCUU
........|......................|.............|.......|
.....((((((((.(((.......))).(((...)))...)))).)))).....  (-14.10)
[0.0, 0.91304347826086951, 1.9130434782608696, 2.0380434782608696]
```

Fig. 7. (Left) Segmentation of the riboswitch with PDB code 3F4H:X. This optimal segmentation has 4 segments, respectively of weights 0.0, 0.913, 1.913, 2.038. Segmentation produced by applying software RNAview,[16] which annotates all hydrogen bonds (canonical base pairs, non-canonical base pairs, single nucleotide stacking). Using the resulting contact map, we determined an optimal segmentation, where the fitness function used involved a weight of 2 for contacts within the same segment and a penalty of 1 for contacts between segments. (Right) Three-dimensional display of the same segmentation, where segments of PDB file 3F4H:X are demarcated in different colors, using Pymol.

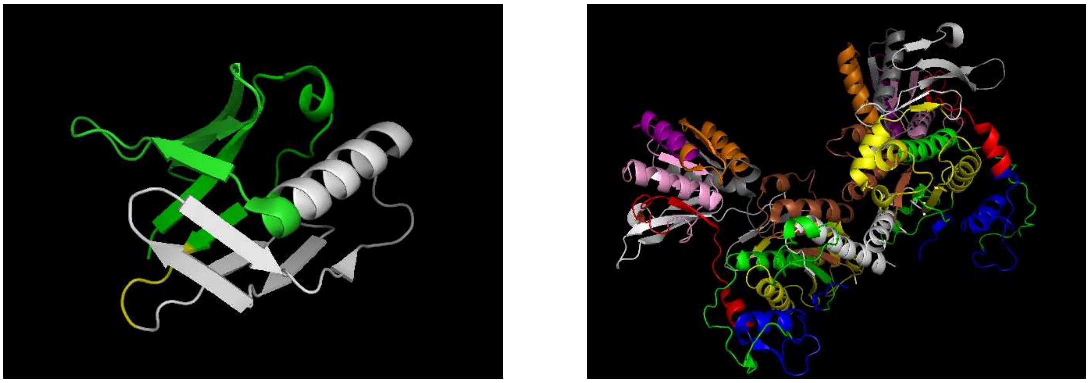


Fig. 8. (Left) Segmentation of the secretin protein with PDB code 1Y9L. Gram-negative pathogens such as Shigella, Salmonella, Yersinia and Pseudomonas use a type III secretion apparatus to translocate virulence proteins into host cells. X-ray structure determined by Lario et al.[7] (Right) Segmentation of the metabotropic glutamate receptor (mGluR) with PDB code 1EWT. X-ray structure determined by Kunishima et al.[6] Each segment in the optimal segmentation is displayed in a different color. Images produced by Pymol.

fitness function whenever X-ray structures are available.

Acknowledgments

We would like to thank Yang Ding for some assistance with scripts, and to Dave Mathews for the reference to the manual segmentations of Jaeger et al.[5] Research of I. Dotú was supported by a grant from Fundacion Caja Madrid, while that of W.A. Lorentz was supported by NSF grants DBI-0543506. Research of P. Van Hentenryck was supported by NSF DMI-0600384. Research of P. Clote was supported by the Foundation Digiteo - Triangle de la Physique, as well as by NSF grants DBI-0543506 and DMS-0817971. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional thanks to the Deutscher Akademischer Austauschdienst for funding a visit of P. Clote to Martin Vingron's group in the Max Planck Institute of Molecular Genetics.

References

1. C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 22(4):469–483, 1996.
2. P. Benaola-Galván, R. Román-Roldán, and J. L. Oliver. Compositional segmentation and long-range fractal correlations in DNA sequences. *Physical Review E*, 53:5181–5189, 1996.
3. P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction.* John Wiley & Sons, 2000. 279 pages.
4. A. V. Finkelstein and M. A. Roytberg. Computation of biopolymers: a general approach to different problems. *Biosystems*, 30(1-3):1–19, 1993.
5. J. A. Jaeger, D. H. Turner, and M. Zuker. Improved predictions of secondary structures for RNA. *Proc. Natl. Acad. Sci. U.S.A.*, 86(20):7706–7710, October 1989.
6. N. Kunishima, Y. Shimada, Y. Tsuji, T. Sato, M. Yamamoto, T. Kumasaka, S. Nakanishi, H. Jingami, and K. Morikawa. Structural basis of glutamate recognition by a dimeric metabotropic glutamate receptor. *Nature*, 407(6807):971–977, October 2000.
7. P. I. Lario, R. A. Pfuetzner, E. A. Frey, L. Creagh, C. Haynes, A. T. Maurelli, and N. C. Strynadka. Structure and biochemical analysis of a secretin pilot protein. *EMBO J.*, 24(6):1111–1121, March 2005.
8. W. Li, P. Bernaola-Galvan, P. Carpena, and J. L. Oliver. Isochores merit the prefix 'iso'. *Comput. Biol. Chem.*, 27(1):5–10, February 2003.
9. W. Li, P. Bernaola-Galvan, F. Haghighi, and I. Grosse. Applications of recursive segmentation to the analysis of DNA sequences. *Comput. Chem.*, 26(5):491–510, July 2002.
10. W. Li, G. Stolovitzky, P. Bernaola-Galvan, and J. L. Oliver. Compositional heterogeneity within, and uniformity between, DNA sequences of yeast chromosomes. *Genome Res.*, 8(9):916–928, September 1998.
11. R. Nussinov and A. B. Jacobson. Fast algorithm for predicting the secondary structure of single stranded RNA. *Proceedings of the National Academy of Sciences, USA*, 77(11):6309–6313, 1980.
12. J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993.
13. R. Román-Roldán, P. Benaola-Galván, and J. L. Oliver. Sequence compositional complexity of DNA through an entropic segmentation method. *Physical Review Letters*, 80(6):1344–1347, February 1998.
14. S. C. Schmidler, J. S. Liu, and D. L. Brutlag. Bayesian segmentation of protein secondary structure. *J. Comput. Biol.*, 7(1-2):233–248, Feb-Apr 2000.
15. S. R. Sunyaev, G. A. Bogopolsky, N. V. Oleynikova, P. K. Vlasov, A. V. Finkelstein, and M. A. Roytberg. From analysis of protein structural alignments toward a novel approach to align protein sequences. *Proteins.*, 54(3):569–582, February 2004.
16. H. Yang, F. Jossinet, N. Leontis, L. Chen, J. Westbrook, H.M. Berman, and E. Westhof. Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Res.*, 31(13):3450–3560, 2003.