k-Local Folding

A Local Alignment Approach to RNA folding

Ben Chugg, Coulter Beeson, Kenny Drabble, Jeff Jeyachandren

The University of British Columbia

April 6,2017

RNA Folding

RNA consists of the four base pairs Adenine (A), Guanine (G) , Cytosine (C) and Uracil (U). These base pairs of RNA pair in a complementary fashion: Adenine to Uracil (A – U) and Cytosine to Guanine (C – G).

Unlike DNA for which we are concerned with optimally aligning two strands, for RNA we are concerned with how the strand folds with itself.

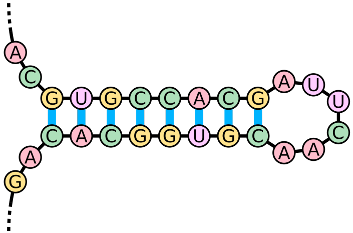


Figure: Source: http://rosalind.info/problems/pmch/

RNA Folding

There are several frameworks with which we can model RNA folding. We will use the Energy Minimization Model.

In this model, matches are scored as +1 and non-matches as 0. This lends itself to a dynamic programming algorithm: Nussinov's algorithm which we will call Nussinov.

RNA Folding

Let $r = r_1, \ldots, r_n$ be a strand of RNA, where $r_i \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$ and let $S(i,j)$ denote the optimal score of folding the subsequence $s_i, s_{i+1} \ldots s_j \subset s$. Then,

$$S(i,j) = \max \begin{cases} S(i+1, j-1) + 1, & \text{if } i,j \text{ base pair} \\ S(i+1, j), & \\ S(i, j-1), & \\ \max\limits_{i<k<j} \{S(i,k) + S(k+1, j)\}, & \text{bifurcation.} \end{cases}$$

It's clear that the runtime is $O(n^3)$.

k-Local Folding

Nussinov returns the mathematically optimal alignment under the energy minimization model. Therefore, any new approach cannot hope to beat the scores, but only improve the running time.

Idea: Subsequences of an RNA strand which are (near) palindromes of each other are likely to be a good match. Pair these regions and pass the leftover segments to Nussinov.

k-Local Folding

Formal Goal: We propose a heuristic based approach to speed up Nussinov with a fast preprocessing step.

Formally, we define an algorithm k-Local Folding which takes as input an RNA strand r and a parameter k, runs a local alignment algorithm on the strand to find k high scoring — and disjoint — palindromic regions of r. It then passes the remaining unpaired regions to Nussinov to be folded as usual.

k-Local Folding

```
procedure k-Local Folding($(r, k)$)
    Initialize stack $S \leftarrow r$
    while The number of local alignments found is $\leq k$ and $S$ not empty do
        $s \leftarrow \mathrm{pop}(S)$
        Let $\overline{s}$ be the reverse of $s$
        Call Local Alignment on $s$ and $\overline{s}$.
        if Local alignment found then
            Remove the aligned regions from $s$
            Push all unmatched regions of $s$ back onto stack
        end if
    end while
    Call Nussinov on all unmatched regions of $r$.
end procedure
```

k-Local Folding

example

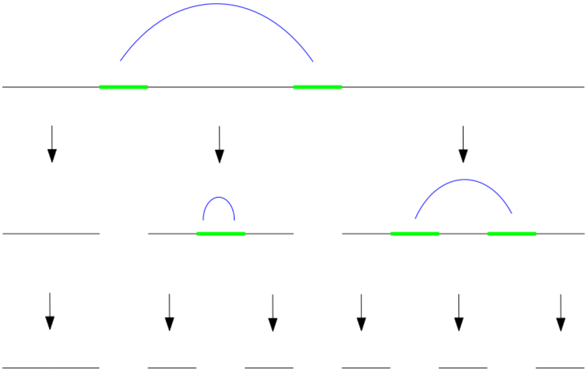


Figure: A strand of RNA undergoing multiple local alignments on successive subsequences. The unmatched regions at the bottom will be passed into Nussinov

k-Local Folding: Runtime Analysis

Fix an RNA strand r with length n.

Lemma (1)

Let $\mathscr{A}$ be the set of local alignments found. Finding k disjoint local alignments of r takes time $O(n^2k)$.

Lemma (2)

k-Local Folding *runs in time*

$$O\left(\left(n - \sum_{A \in \mathscr{A}} \ell_A\right)^3 + n^2k\right),$$

where ℓ_A is the length of an alignment $A \in \mathscr{A}$.

Proof of Lemma 1.

We can view the progress of the algorithm as a ternary tree: amortized across each level we perform local alignment on a sequence of size $n - \sum_{A \in \mathscr{A}} \ell_A$. Local alignment takes time squared in the size of the input. Since the tree in the worst case has depth k, the result follows. ◀

Proof of Lemma 2.

Running Nussinov takes cubic time in the size of the input. k-Local Folding first finds k disjoint alignments, then runs Nussinov on the remaining unmatched regions, which have total length $n - \sum_{A \in \mathscr{A}} \ell_A$. The result follows by applying lemma 1. ◀

How big is the tree?

remark

It's a very unlikely case that the depth of the tree is k: it will more likely be $\log(k)$. Therefore, an average case analysis will yield that k-Local Folding runs in time

$$O\left(\left(n-\sum_{A\in\mathscr{A}}\ell_A\right)^3+n^2\log(k)\right).$$

k-Local Folding: Runtime Analysis

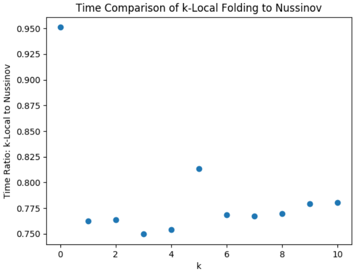
We remark that since $\sum_{A\in\ell_A}\ell_A^3 \in O(n^3)$, $O((n-\sum_{A\in\mathscr{A}}\ell_A)^3) = O(n^3)$ so there is no asymptotic difference between k-Local Folding and Nussinov. However, our hypothesis was that there may a difference in the run times in practice.

example

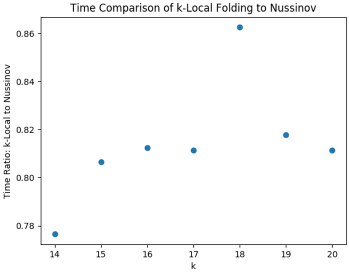
In the extreme case, suppose r is a perfect palindrome. Then we do $O(n^2)$ work instead of $O(n^3)$, so we gain a factor of n.

Results: Runtimes

Each trial was run with 20 random sequences, and the results were taken as the average of those trials. The following data is for 16S Ribosomal Subunit RNA.



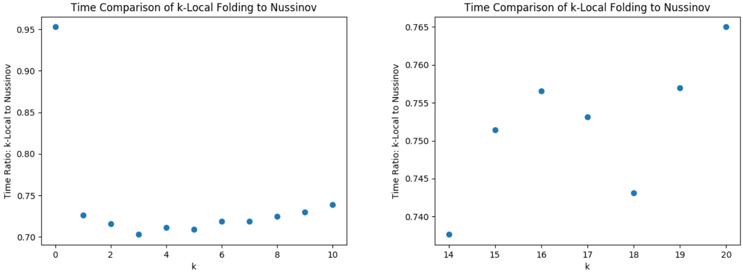
(a) $k \in \{0, 1, \ldots, 10\}$



(b) $k \in \{14, \ldots, 20\}$

Results: Runtimes

Ciliate Telomerase RNA data.

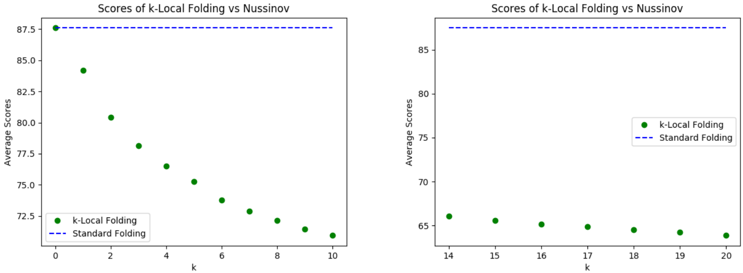


(c) $k \in \{0, \ldots, 10\}$

(d) $k \in \{14, \ldots, 20\}$

Results: Scores

16s Ribosomal Subunit RNA

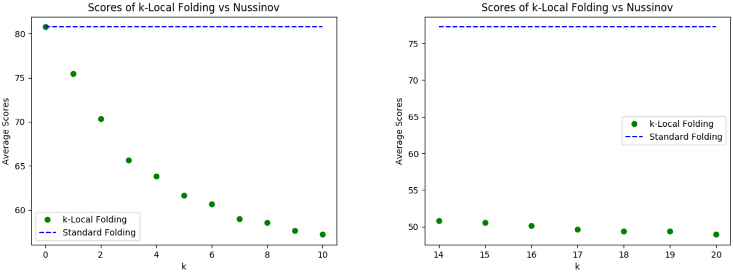


(e) $k \in \{0, \ldots, 10\}$

(f) $k \in \{14, \ldots, 20\}$

Results: Runtimes

Ciliate Telomerase RNA data.



(g) $k \in \{0, \ldots, 10\}$ (h) $k \in \{14, \ldots, 20\}$

Concluding Remarks

1. Results look fairly invariant under different data.
2. k-Local Folding may be best used as a preprocessing step for traditonal RNA folding algorithms for regions which are likely hairpinned.

Extensions and Further Research

1. Can we speed up k-Local Folding by being smarter with our data structures? Tempting to only do local alignment once ...
2. Modify k-Local Folding to report possible pseudoknots
3. Implement k-Local Folding to use the Four Russians speedup of Nussinov.
4. Probabilistic (Viterbi-like) Approach
5. Providing better bounds on the runtime based on the expected number of palindromic like regions found on an alignment.

The code and slides can be found at `https://github.com/bchugg/bwt`.