Structure-Informed RNA Sequence Alignment using Discriminative Models

Group: Gregory Goldgof
Advised by Chuong Do and Serafim Batzoglou
Date: November 15, 2006

Overview

RNA is a nucleotide polymer transcribed from DNA. Once thought of as only a messenger molecule, RNA is know recognized to be essential to a wide range of cellular processes including transcription, translation, and gene regulation. RNA sequence alignment has applications in RNA structure prediction, phylogeny building, and the detection of unknown function non-coding RNA (ncRNA) sequences in the genome. Consequently, accurate RNA sequence alignment is an essential tool needed to understand basic biological and evolutionary processes.

RNA sequence alignment remains a challenge for computational biologists since ncRNA can evolve by compensatory mutations, which maintain nucleotide base pairings, but mask sequence homology. An RNA molecule's pattern of base pairings is called its 2D structure, or folding (*figure1*). Simultaneous sequence alignment and structural alignment leads to more accurate alignments because both structure and sequence are evolutionarily conserved at some rate. The superiority of this method was demonstrated by the Sankoff algorithm which simultaneously predicts RNA sequence alignment and 2D structure, yielding higher quality alignments than previous algorithms. The downside of this algorithm is that in runs in $O(n^3)$ time with respect to sequence length making it useless for many common alignment applications.

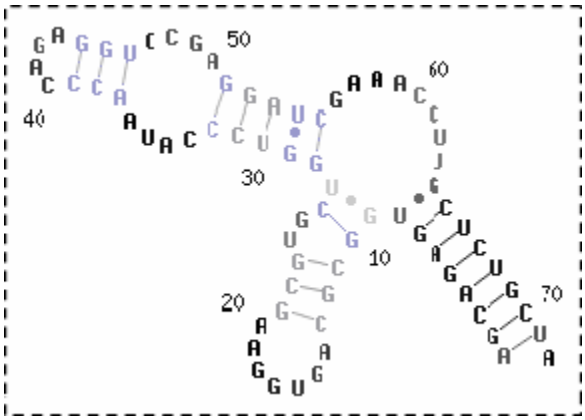

Figure 1: RNA 2D structure or folding is the pattern of nucleotide base pairings.

As a compromise, the StrAl algorithm proposed in 2006 performs RNA alignment using a condensed representation of RNA 2D structure. It performs the standard $O(n^2)$ algorithm for sequence alignment, however the scoring function takes into account

sequence similarity as well as up-stream and downstream pairing probabilities. A weakness of the algorithm is that the relative importance of structural versus sequence alignment is hand-tuned by the authors.

Supervised machine learning can be used to improve the performance of structurally informed $O(n^2)$ RNA alignment algorithms such as StrAl. The parameters that determine the relative importance of structure and sequence can be optimized, as well as the parameters for the sequence substitution matrix. The goal of this project is to develop the best performing quadratic time RNA alignment program. This work will hopefully lead the development of higher quality RNA structural prediction, phylogeny building and gene finding.

Methods:

Step 1: **COMPLETE** Implement the Viterbi algorithm for finding the optimal alignment based on the parameter set. The program uses a modified Needleman-Wunsch algorithm for global sequence (string) alignment, with affine gap-penalties. This program is based on a dynamic programming algorithm that finds the best scoring sequence (Viterbi parse) based on a gap-opening penalty, gap-extension penalty, and nucleotide substitution matrix. The dynamic programming is based on the following recursions:

$$\mathbf{M}(i, j) = \min \begin{cases} \mathbf{M}(i-1, j-1) + s(\mathbf{x}_i, \mathbf{y}_j) \\ \mathbf{I}(i-1, j-1) + s(\mathbf{x}_i, \mathbf{y}_j) \end{cases}$$

$$\mathbf{I}(i, j) = \min \begin{cases} \mathbf{M}(i, j-1) - d, \\ \mathbf{I}(i, j-1) - e, \\ \mathbf{M}(i-1, j) - d, \\ \mathbf{I}(i-1, j) - e, \end{cases}$$

In the above recursions M(i, j) is the match score at position (i, j), I is the insertion score at position (i, j), $s(x_i, y_j)$ is a score for substituting the nucleotide at position i of sequence 1 with the nucleotide at position j of sequence 2 (the value from the sequence substitution matrix), *d* is the gap opening penalty and *e* is the gap extension penalty.

The alignment is then reconstructed based on pointers from the dynamic programming matrices. An example matrix and constructed alignment:

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	2	2	2	2
A	0	1	2	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3	3
C	0	1	2	2	3	3	4	4	4	4	4	4
G	0	1	2	2	3	3	4	4	5	5	5	5
A	0	1	2	3	3	3	4	5	5	5	5	6

Result

GA-ATTCAGTTA
G-A-T-C-G--A

Step 2: **COMPLETE.** Learn nucleotide substitution matrix, gap-opening penalty and gap extension penalty using the Perceptron algorithm. The highest likelihood parse is the result of Step 1. Correct alignments are taken from the hand-curated Rfam database of RNA family alignments. This step also involves creating a representative training set based on a random sapling of the hand-curated alignments in Rfam.

Step 3: **COMPLETE.** Add structural information into the Viterbi algorithm. The program will use CONTRAfold, a probabilistic RNA folding algorithm to fold each of the sequences. It will then represent the output of CONTRAfold as a matrix of base-pairing probability vectors as described in the StrAl paper.

Step 4: **In Progress.** Learn the structural substitution matrix and structure/sequence tradeoff parameters. The Perceptron-based algorithm created in Step 2 will be modified to include the structural information from Step 3.

Step 5: **In Progress.** Experiment with different feature representations. The length and extant of this section will be determined by time constraints and may be done after the conclusion of the course.

Step 6: Perform formal testing. Benchmark performance of developed program against StrAl, Clustal, Sankoff-based algorithms and others.

Conclusion

The program, starting from randomly assigned feature weights, was able to learn parameters for making accurate sequence alignments. The weights reflected known evolutionary phenomenon. More specifically, matches were rewarded, whereas mismatches and gaps were negatively weighted. Unfortunately, there is nothing to benchmark this program against since modern sequence aligners use richer feature sets than the one currently used by the program. Furthermore, many of them, such as CONTRAlign do you use machine learning approaches to optimize their feature weights, so there is no reason to expect superior performance from the developed program.

The integration of structural elements into the feature vector is currently under development. The Perceptron algorithm should learn these feature weights just as they learned the weights of features derived from sequence elements.

Discussion

I plan to continue this work throughout the next quarter. First of all, I am interested to see how much of an advantage the introduction of structural features can confer on alignments. In addition, experimenting with different feature representations of the structure as well as seeing the consequent change in learned weights may provide insight into the functionally aspects of RNA secondary structure. For example, it may demonstrate which features are important and which features overlap. I also hope to experiment with different machine learning algorithms to see which one works best for this particular problem. Hopefully, implementation of a wide range of these algorithms in the same setting will give me a better understanding of their differences in terms of implementation complexity, accuracy for this type of problem, and specific run-time. Finally, I hope the project will demonstrate the superiority of the learned approach over the approach utilized by StrAl. It is my opinion that a machine learning approach will allow me to integrate and optimize a more complex feature set than StrAl's, leading to better alignments. I hope that the final product will be of direct use to biologists and be integrated into future bio-computation tools.

Results Some sample predicted alignments are included.

Sample Alignment 1, Sequence 1:
GUCCCUAACUAGA

Sample Alignment 1, Sequence 2:
.UCCC...CUGGA

Sample Alignment 2, Sequence 1:
GGGUCCUAAAGUGGGCUACUGUGAGUCCCUAACUAG.AGCUACUUUUUGUCGGGCGAGU
CCCUAACUAGAU
CC.C...CUG.GA..UCCCCUGGA

Sample Alignment 2, Sequence 2:
.....AAAUUGG..UGAUGU.A.UC....AUUAGUAUCCCCUGGAGGG.GGCCUUUU
CCC...CUGGAUCCACACACGGUGACGUACCCUGGA

Sample Alignment 3, Sequence 1:
AGGU.....GUGAUGUAUGCUUCGUGUAGUCUAGUUAGUUAGUUGACGCGUGUG
UCGUGUCAU.GAUUAUGUAGAUCUAGAUGUCUGUCGAUGUAUCGAUUCAACUGUCGUUG
CGUCGUUGGU..CGUAUCACAACACUGUCGUCGUAUAAACACUAUAUUUUUUUUGAUGA
UCA.....AAGCGAAA..UAUAGCUGAUAUAUAGUAUCGUCUAGUCGUAUCGU

Sample Alignment 3, Sequence 2:
AGGUAGGGUGUGCCGUGAUGUGUGACACAUGUAGUCUAGUUAGUUAGUUGACGCGUGUA
AUGA.UGAUCGAUUAUGUAGAUCUAGAUGUCUGUCGAUGUAUCGAUUCAACUGU...UG
CGUCGUUGGUGACGCA..ACAACACUGUCUAUAUAUAAACACUAUAUAUUUUUUGAUGA
UCAGCACCAAGTATAAAAUAUAGCUGAAAUAUAGUAUCGUAGUGUCGUAUCGU

Reference

1. Dalli D, Andreas Wilm, Indra Mainz, Gerhard Steger. *StrAl: progressive alignment of non-coding RNA using base pairing probability vectors in quadratic time*. Structural Bioinformatics, 22(13): 1593-1599, 2006.

2. Durbin R, S Edy, A Krogh, G Mitchison. Biological Sequence Analysis. *Dynamic programming with more complex models*. Cambridge: Cambridge University Press (1998): 28-32.