Apr 17th, 3:00 PM - 4:00 PM

A Convolutional Neural Network to Trim Sequence Alignment Overextension

Jack Roddy

A Convolutional Neural Network to Trim Sequence Alignment Overextension

Jack Roddy and Travis Wheeler, Department of Computer Science, University of Montana

What and why

A key component of modern molecular biology is sequence annotation - labeling the contents of biological sequence. Annotation largely depends on identifying relationships between sequences through the use of sequence alignment. Modern methods for sequence alignment are remarkably good at recognizing when a substring of one sequence is related (aligns) to a substring of another sequence, but are also prone to a form of error known as **alignment overextension**, in which the alignment extends beyond the true bounds of relatedness. The impact of overextension is substantial - for example, in the annotation of transposable elements in the human genome, we have estimated that 2% of the annotated genome (~30 million nucleotides!) is the result of overextension. Current methods used to combat overextension are only somewhat effective, and can have the unintended consequence of reducing search sensitivity and over-trimming the alignment. We developed Machine Learning approaches to identify and trim overextended regions in sequence alignments. We benchmark the trimming using an artificial sequence dataset that mimics transposable elements inserted into simulated sequence alignment. Our results demonstrate a dramatic decrease in overextension with a minimal amount of over-trimming.

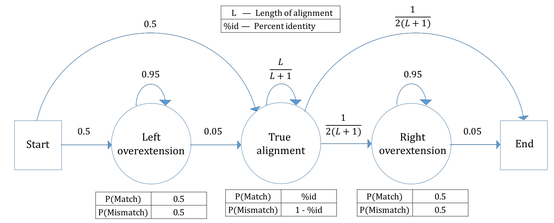
```
actaaaaatacaaaaaattagccgggcgtggtgg..cgggcgcctgtagtcccagctactcgggaggctgaggcaggagaatggcgtgaacccgggaggcggagcttgcagtgagcc
 || | | || ||  |||| |     |  |   ||  |||| |||||||||||||||||| ||||||||||||||||||||||||||||||||||||||||||||||| |||||||| ||||||||||||||||
TCTCACAGTATAATCAATTTGGAAAGTTTCTTTGaaCGGGGGCCTGTAGTCCCAGCTTCTCGGGAGGCTGAGGCAGGAGAATGGCGTGAACCTGGGAGGCAGAGCTTGCAGTGAGCC
```

Overextension — True alignment

Methods

- Generate a sequence that accurately simulates a human genome with inserted transposable elements (TE) (using the GARLIC[1] algorithm)
- Produce a sequence alignment for each inserted TE and measure exactly how much each alignment was overextended (using HMMER[2], which already implements state of the art methods to combat overextension)
- Trim alignments with a Hidden Markov Model (HMM)
- Train a CNN to classify overextended regions
- Apply the CNN to trim alignments that were not used for training

HMM trimming

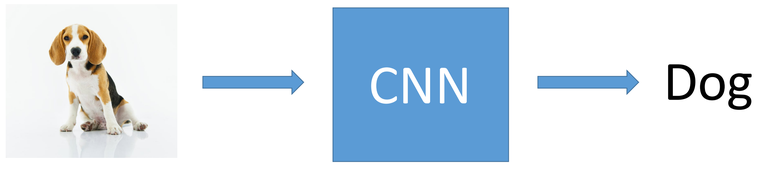


A Hidden Markov Model (HMM) is a probabilistic generative model that can be used to label the unobserved processes producing observable data. In this case, we model sequences as being created from three states: (i) "left overextension" and "right overextension" states that produce alignment columns of low percent identity, and (ii) a "true alignment" state that produces alignment columns of relatively high percent identity. The observed alignment is imagined as being produced by a passage through these states, and we aim to identify the passage with highest probability.

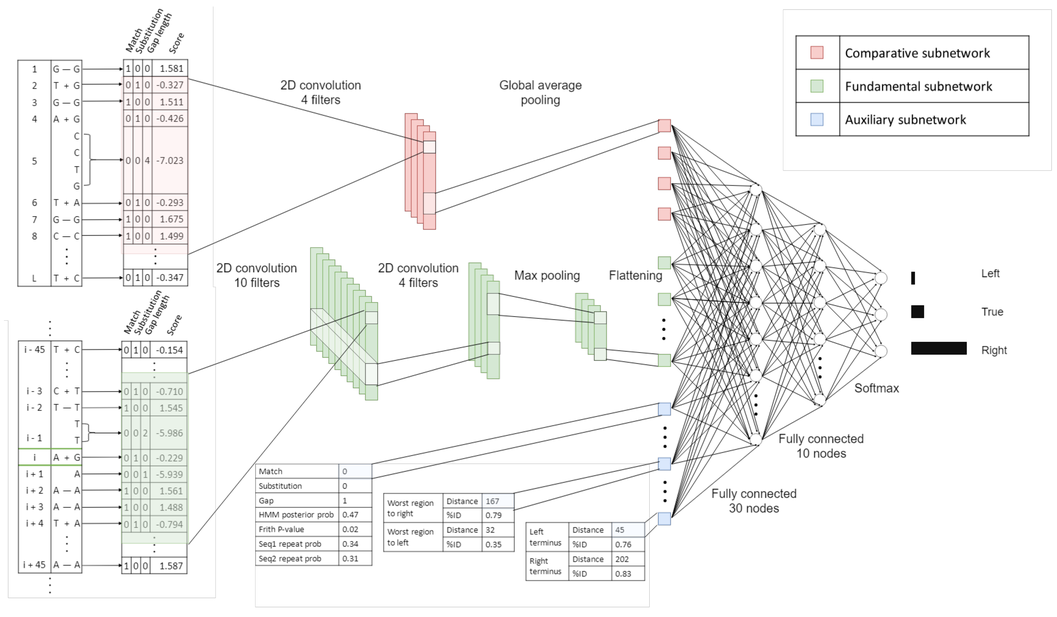
What is a CNN?

A convolutional neural network, or CNN, is a Machine Learning construct that generally aims to apply a label to each individual element in a set of input data. CNNs are most frequently applied to the task of recognizing and labeling the contents of an image.

CNNs are trained with datasets in which each element has already been correctly labeled. During the training process, the CNN will make a prediction of the label for each example in the training dataset. Using a metric that quantifies the amount of misfit between the predicted label and the true label, various parameters in the CNN are adjusted. As this process is repeated, the classification accuracy of the CNN increases.



CNN trimming



We developed a convolutional neural network to classify overextended regions in sequence alignments. For a specific position in an alignment, a collection of features are fed through the network to predict whether that position is a left overextension, right overextension or part of the true alignment.

Our CNN architecture has three subnetworks:

- The fundamental subnetwork
- The comparative subnetwork
- The auxiliary subnetwork

The fundamental subnetwork applies 2D convolution on local information around a position in an alignment. The comparative subnetwork utilizes 2D convolution and global average pooling (GAP) to perform extreme dimensionality reduction on information about the entire alignment. The auxiliary subnetwork adds supplementary information about the position.

The information from each of these subnetworks is combined in the final, fully connected layers to produce a probability of the position belonging to each of the three classes.

Examples of trimmed alignments

———	True alignment
———	HMM trimmed
———	CNN trimmed

```
taactggttg....aacaactatacccaaa.aagtgttgattaatggattaatgtcaacctggagggaggtctctagtggtatgccacagggctctgtccttggccc
|   ||||||    || ||   | |||||||  ||||  ||| |||||||||||||    ||  |               |  ||||  |      |  |||||  ||||
TGCTTGGTTGgggcAAAAAT--TGCCCAAAaGAGTGCAGAT-AATGGATTAATGTGGTTCTATA-------------GAAATGCTA-----TTTCGTCCTAGGCCT

tcctggctctgccact.tactagctgtg.tgaccttgggcaagttacttaacctctctg.agcctcagtttcct..c..atctgt.aaaatggg.gataataatacctacctcatagggttgtt...gtgaggattaaatga
|  ||||| ||||| | | | ||| | |   || | |||| ||||| ||||| || | || |||||||| ||  |   |  ||  |||||||| || || ||| || |   ||||  |      |  |||||||| ||
TTTTGGCTTTGCCATTaTGCCAGCAGAGaAGATCCTGGGGAAGTTTCTTAATCTTTTTGtAGCCTCAGGTTTTTttCgaTTTTGAaAAAATGGGgGAAAAAAATGCCCAT-TTGCAGGGAAGAAgaaGAAGGGATTAAAAGA

tccatggcc.agaga..ctcaaaaagaaagatggctcaagagggttgggaagctcttaaaaatgaaattctgattacaattacaaatgatccaatgag.gaagaaaagagggagggcatttaagaaaccaatatggcttcata
| || |||  ||||   || || |  ||| | |||| |||||||        || |||||||   |||| ||| |  | ||  | |  ||  || ||| |||||||||| |||||||||| ||||||  ||||| |||| |
TTCAAGGCTgAGAGGagCTTAATAGAAAACAAGGCT-AAGAGGGCG---GGACTTTTAAAAAG--AATTTTGAATTTATTT-TA-AAAATTTAAAGAGaGAAGAAAAG-GGGAGGGCAT-TAAGAAGCCAAT-TGGCCTT---

tagtcctact.g..tactctgcactggtcagacca.catctgga..gtattgtgtt.cagttctggg.taccacattttaagagggaca...ttgacaaa..ctggagtatatccagaggagagtga...ccaggat.ggtga
 |||  |   |  |  | ||||| | | || ||| || |  |  |||  || ||||| |||| | |   |||||||| || |     ||||||||  | |||| || | ||||||||||  |  ||||||  || ||
AAGTATTTTCaGaaTTGTTGCCCCTGGTTAAAACAcCATTTGCAaaGATTTGCATTaCAGTTTTGGGaTTCTTTATTTTAAGAAGGGCTattTTGACAAAgaCAGGAGCATGTTCAGAGGAGAGAAAgtgCCAGGAAaGGAGA
```

Summary results

Our results are split into two categories. One groups alignments by overall percent identity and the other by length of overextension.

For each category, we measure the results of each trimming method by the amount of overextension remaining and the amount of under extension (trimming too far) in each alignment.

% Identity	< 60% n = 1626		60% - 70% n = 10967		70% - 80% n = 5011		80% - 90% n = 1249		> 90% n = 232		Overall n = 19085	
Trimming approach	Over	Under	Over	Under	Over	Under	Over	Under	Over	Under	Over	Under
None	13.5	-	6.5	-	2.4	-	2.2	-	1.6	-	5.7	-
HMM	10.2	1.0	2.4	2.4	0.5	4.1	0.3	1.6	0.3	1.2	2.4	2.6
CNN	5.7	2.7	2.1	2.4	1.0	2.4	1.0	0.2	1.6	1.1	2.0	2.3

Input overext length	0 – 10 n = 16338		10 – 20 n = 1297		20 – 40 n = 861		40 – 60 n = 288		> 60 n = 301		Overall n = 19085	
Trimming approach	Over	Under	Over	Under	Over	Under	Over	Under	Over	Under	Over	Under
None	1.4	-	13.7	-	27.2	-	47.7	-	100.3	-	5.7	-
HMM	0.6	2.6	5.6	2.7	11.5	2.8	18.3	4.5	46.7	4.5	2.4	2.6
CNN	0.8	2.3	3.5	2.1	6.5	1.6	12.9	1.4	37.5	2.1	2.0	2.3

[1] Juan Caballero, Arian F. A. Smit, Leroy Hood, Gustavo Glusman. Realistic artificial DNA sequences as negative controls for computational genomics, *Nucleic Acids Research*, Volume 42, Issue 12. 2014
[2] Travis J. Wheeler, Sean R. Eddy. nhmmer: DNA homology search with profile HMMs, *Bioinformatics*, Volume 29, Issue 19. 2013

NIH, P20GM103546 (courtesy of a CBSD Junior Investigator grant).