Joshua Burkhart

12/11/2015

BMI 650

# Final Project Writeup

## Aligner Choice

Considering our assignment involving RNA-Seq alignment, an evaluation paper[1] was reviewed before choosing STAR[2]. STAR was commended in the evaluation[1] for its execution speed, far lower than other RNA-Seq aligners evaluated; its ability to use annotation information to detect splice sites; and its high accuracy.

The package was built from source on church.ohsu.edu. As church.ohsu.edu runs g++ 4.6.3 (which doesn't support c++11), the STAR/source/Makefile had to be modified; -std=c++11 was replaced with -std=c++0x. The reported version is STAR\_2.5.0b\_modified.

## Naïve Approach

### Summary

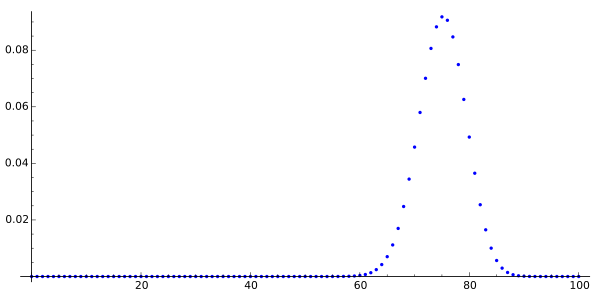
This approach uses combinatorics and information theory to estimate the likelihood of a read aligning to a particular position in the reference.

Assuming a read length of 100, a uniform random (UR) distribution of nucleotides A, T, C, and G, in a read, and a reference sequence with minimum entropy, a read will align without mismatches to a reference with a probability of

This is an acceptably low probability. Using a *good* value for acceptable probability, such as .005, will enable us to allow many more mismatches. First, lets see how to calculate the probability that a UR read will align with exactly one mismatch.

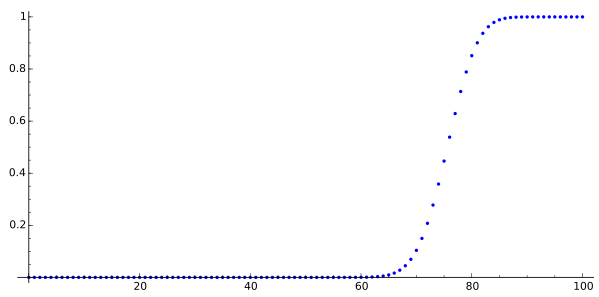
Here, we're multiplying the chance of 99 nucleotides matching by the chance of one nucleotide not matching by the number of ways one mismatched nucleotide can be chosen from a read length of 100. In general, we can use this form to derive a probability mass function (pmf) as below.

Where *l* = the read length, *m* is the number of mismatches, and *m* <= *l*. The pmf for *l* = *m* = 100 is plotted[a] below. *m* is on the horizontal axis and probability is on the vertical axis.



But we're trying to find a threshold, thus we're interested in understanding the probability that a UR read will align with *m or fewer* mismatches. For this we compute the cumulative mass function (cmf) as below.

As above, the cmf for *l* = *m* = 100 is plotted[b] below. *m* is on the horizontal axis and probability is on the vertical axis.



Judging from the cmf plot above, we can see that the highest value of *m* that will give us a probability of a UR read aligning < .005 is somewhere around 60; in fact, for *l* = 100, it's 62.

This can be considered the absolute upper bound of our mismatch threshold. There are still two glaring issues:

1. We're not aligning one UR read. We're aligning 17,220,350[c].
2. We're not drawing from a UR.
3. We don't have

First, we can adjust our cmf to compensate for the number of reads as below.

Neither Wolfram Alpha[3] nor Sage Math Cloud[4] produce plots for this cmf, though Wolfram Alpha[3] does produce a value, allowing us to calculate the highest value of *m* that will give us a probability of all our 17,220,350[c] reads aligning with a probability <.005: 45.

Second, we can adjust our cmf to account for the actual distribution of nucleotides in our reads.

xxx

The software developed for this assignment[5] calculates the following ratios for the RNA-Seq reads.

xxx

Last, we can adjust our cmf using a 0-order Markov Model to account for the actual entropy in our reference.

xxx

The software developed for this assignment[5] calculates the following ratios for the reference transcripts.

A:0.28908129592053294

T:0.28797606910667517

C:0.20011402374243129

G:0.19987491778448566

xxx

### Assumptions

One of the benefits of a naïve analytical solution to this problem is that relatively few assumptions are made. Most will involve the behavior of the aligner (e.g. assume it will handle all splice sites equally, assume it will treat all mismatches equally, etc.). I suppose absence of below limitations 1 and 2 could be considered an assumption.

### Limitations

1. Gaps are ignored
2. Ambiguous calls (N's) are ignored
3. Reference entropy could be more accurately calculated using a higher-order Markov Model; this calculation can be time consuming. Obviously, at an *l*-order Markov Model, we'd be better off using a simulation of artificial reads.

### Alignment Percentages

## Strain Specific / General Approach

### Summary

Remembering that the above calculated value can be considered a strong upper bound for this problem, we'll aim to find a lower biologically-motivated threshold by calculating the fewest possible mismatches we can allow without significantly dropping read coverage.

The VCF file shows alternate calls that, if within exons, can be expected to count as mismatches when aligned to the reference.



If some number of alternates, *a*, falls within the length of a read, *l*, at least *a* mismatches must be allowed to occur for an alignment to take place. We'll refer to any region of length *l* with positive *a* as alternate-dense.



Remembering that our reads were generated using RNA-Seq, we must consider the stitching process that occurs following transcription whereby distal exons can become adjacent in a transcript prior to assessing alternate-density.



The software

Assuming a 0.1% error rate[6] (and that said error rate is UR), we can estimate the ratio of reads that whose alignment to our alternate-dense region would be rejected despite being that region being its biological origin.

Disregarding 7.8% of the reads in this region seems a bit high; allowing for an additional mismatch will allow us to retain many more.

Disregarding only .6% of our reads seems far more appropriate, especially considering that the region is already known to be alternate-dense. It's possible some biological mechanism is in place that increases the likelihood of point mutations.

We'll allow for a single additional mismatch to account for both the known sequencing error rate[6] and the likelihood of a point mutation, increasing our threshold to xxx

### Assumptions

### Limitations

### Alignment Percentages

## References

[1] Engström, Pär G., et al. "Systematic evaluation of spliced alignment programs for RNA-seq data." *Nature methods* 10.12 (2013): 1185-1191.

[2] Dobin, Alexander, et al. "STAR: ultrafast universal RNA-seq aligner." *Bioinformatics* 29.1 (2013): 15-21.

[3] <http://www.wolframalpha.com/>

[4] <https://cloud.sagemath.com/>

[5] <https://github.com/joshuaburkhart/BioinformaticsAlgorithms/tree/master/final_proj>

[6] Christina Zheng, zheng@ohsu.edu

## Notes

[a] plotted using the Sage Math Cloud[4] with the python code below.

pmf=lambda m: ([.25^(100-m) \* .75^(m) \* binomial(100,m)])

pts = [point([i,pmf(i)[0]]) for i in range(0,101)]

p = add(pts)

show(p)

[b] plotted using the Sage Math Cloud[4] with the python code below.

cmf=lambda x: add([.25^(100-x) \* .75^(x) \* binomial(100,x) for x in range(0, x)])

pts2 = [point([j,cmf(j)]) for j in range(0,101)]

p2 = add(pts2)

show(p2)

[c] calculated from the data given for this assignment with the below command.

$ cat PWK\_R1.fastq | grep '>' | wc -l

17220350