CSE 549 - Project Midterm Report

Implementing Long Read Assembly Algorithms - Minhash and Containment Hash

Prashanth Soundarapandian(111498721), Shalini Bhaskara(111500114),
Laxmi Prashanthi(111401549), Naga Srinikhil Reddy(111461912)

Minhash

MinHash reduces a string (broken down into multiple k-mers) to a small set of fingerprints, called a sketch. We then compute the Hamming distance between the sketches of two k-mer sets to find their Jaccard similarity.

To begin with, we split the given strings, *T1* and *T2* to a set of k-mers (also known as shingles) to integer fingerprints using multiple MurmurHash3 functions with differing seed values. For each hash function, only the minimum valued fingerprint, or min-mer, was retained. The collection of min-mers for each of the given strings, *T1, T2* make the sketches *S1, S2*.

Finally, we calculated the Jaccard similarity for the two sketches using the formula:

$$similarity = \frac{intersection(S1, S2)}{union(S1, S2)}$$

Containment Hash

Based on the number of hashes ***(k)***, size of the bloom filter ***(m),*** numbers of elements inserted into the bloom filter ***(n),*** we calculate the false positive error rate ***(p),*** based on the given formula.

$$p \approx (1 - e^{\frac{-kn}{m}})^k$$

In our containment hash implementation, we assumed a permissible false positive error rate***(p)*** and an approx. number of elements that could be inserted into the bloom filter ***(n)*** to arrive at a value for number of hashes ***(k)*** and size of the bloom filter ***(m)***.

$$k = \frac{m}{n} \ln 2 \qquad m = -\frac{n \ln p}{(\ln 2)^2}$$

The hash function used is based on a MurmurHash3 hash and is implemented as shown.

$$hash_i(x, m) = (hash_a(x) + i * hash_b(x)) \% m$$

We create a bloom filter of size***(m)*** bits and apply***(k)*** hashes on all possible k-mers of the large string. Then, we create the minhash sketch of the smaller string and determine the intersection between the minhash sketch of the smaller string and the bloom filter of the larger string to arrive at the containment estimate as shown:

$$containmentEstimate = \frac{intersectionCount}{numOfHashes}$$

Finally, the Jaccard estimate through containment is determined as follows:

$$Jaccard\ Estimate = \frac{a * containmentEstimate}{a + b - a * ContainmentEstimate}$$

a=len(KmersInSmallStr),b=len(KmersInLargeStr)

Observed Jaccard Estimate values using Minhash and Containment hash approaches.

K-mer Size	Num. of Hashes in MinHash	Jaccard Estimate using MinHash	Jaccard Estimate using Containment	Plain Jaccard	Difference Observed for Minhash from True Jaccard	Difference Observed for Containment Hash from True Jaccard
20	5	0.111111	0.0727211	0.0727211	0.0384	0.0000
20	10	0.111111	0.0727211	0.0727211	0.0384	0.0000
20	100	0.0416667	0.0727211	0.0727211	0.0311	0.0000
20	200	0.0498688	0.0727211	0.0727211	0.0229	0.0000
20	300	0.0471204	0.0727211	0.0727211	0.0256	0.0000
20	400	0.0471204	0.0727211	0.0727211	0.0256	0.0000
20	500	0.0471204	0.0727211	0.0727211	0.0256	0.0000
20	600	0.0452962	0.0727211	0.0727211	0.0274	0.0000
20	700	0.0471204	0.0727211	0.0727211	0.0256	0.0000
20	800	0.0471204	0.0727211	0.0727211	0.0256	0.0000

*Sampled on two genomes of length 5877 and 445.