1: simulated PacBio reads

(a)

To estimate a reasonable k, we can follow the method section in MHAP paper. So the expected number of kmer $E[X_c]$ can be calculated use following equation:

$$E[X_c] = (P_{ovl} - P_{ovl}P_{rand})M + P_{rand}L$$

M is the size of overlap part and L is the length of the read. P_{ovl} and P_{rand} are given by following equation:

$$P_{rand} = 1 - (1 - |\Sigma|^{-k})^K$$

$$P_{ovl} = [(1-\varepsilon)^2 + \varepsilon^2 \frac{1}{|\Sigma| - 1}]^k$$

K is the number of kmers in the read, we can assume is equal to the length of read, ε is the error rates.

From this calculation, we found that when $k = 16$, usually we can only expect several kmers. For our MinHash sampling process,I think the k = 16 is too large. I tried with $k = 10$ and $k = 12$, as the expected number of matches is much higher.

For number of permutations, I believed the reasonable value should be proportional to the read length. Given the thousands of bases in Pacbio Reads, I think $m = 512$ and $m = 1024$ are reasonable choices.

I was trying to follow the code that use false positive rate and false negative rate to optimize the number of r, but it always returns 1, so right now in my experiment the r always be 1.

Jaccardi similarity threhold is determined by a small scale test, I found for most overlapped reads, the similarity for 10-mer is only 0.01-0.10 level. So I choose 0.01 as my threshold. I also use 0.04 threshold which is given from MHAP paper

For 30X dataset:

1. k=10 1024 0.01
   116444191 pairs passed the filters
   sensitivity 0.9782145414156769
   FPR 0.8314137831129967
   accuracy 0.0040011012657557125
   F1 0.007969605185993432

2. k=10 1024 0.04 9722271 pairs passed the filters
   sensitivity 0.4537027511070146
   FPR 0.06814697975590785
   accuracy 0.022226288487535474
   F1 0.04237660405124178

3. k = 12 1024 0.01
   26040512 pairs passed the filters sensitivity 0.8414108477978336
   FPR 0.18380383018735438
   accuracy 0.015389405553930736
   F1 0.03022597793028742

For 20X dataset:

1. k=10 1024 0.04
   5201162 pairs passed the filters
   sensitivity 0.3249885241959911
   FPR 0.008208450660925973
   accuracy 0.04001951871524094
   F1 0.07126354927725775

2. k = 12 1024 0.01
   38755561 pairs passed the filters
   sensitivity 0.7990532071359203
   FPR 0.06287228961496219
   accuracy 0.013205227502705998
   F1 0.025981089114060423

From the above result, I believed k = 12 with 0.01 threshold is more reasonable for our filter task. k= 10 with 0.01 threshold has too many false positive pass our filter. And k = 10 with 0.04 has a lot of false negative situations.

(b)

For the memory usage (30X dataset):

k	m	threshold	memory	running time
10	1024	0.04	3354296kb	03:10:37
10	512	0.01	832444kb	03:16:16
10	1024	0.01	1235856kb	03:16:26
12	1024	0.01	1401860kb	03:51:17

(c)

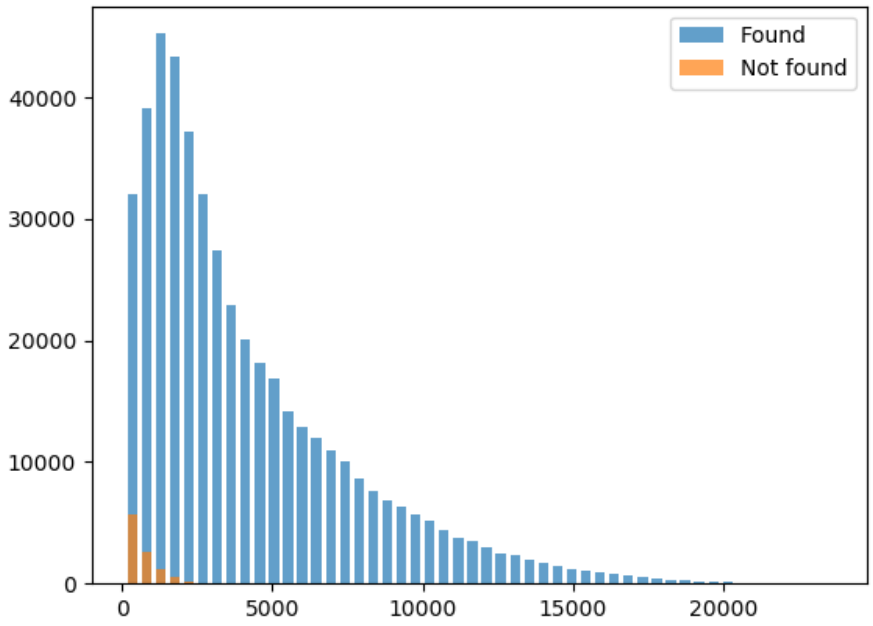

Figure 1: Overlap size distribution for k =10 with 1024 permutations and 0.01 threshold

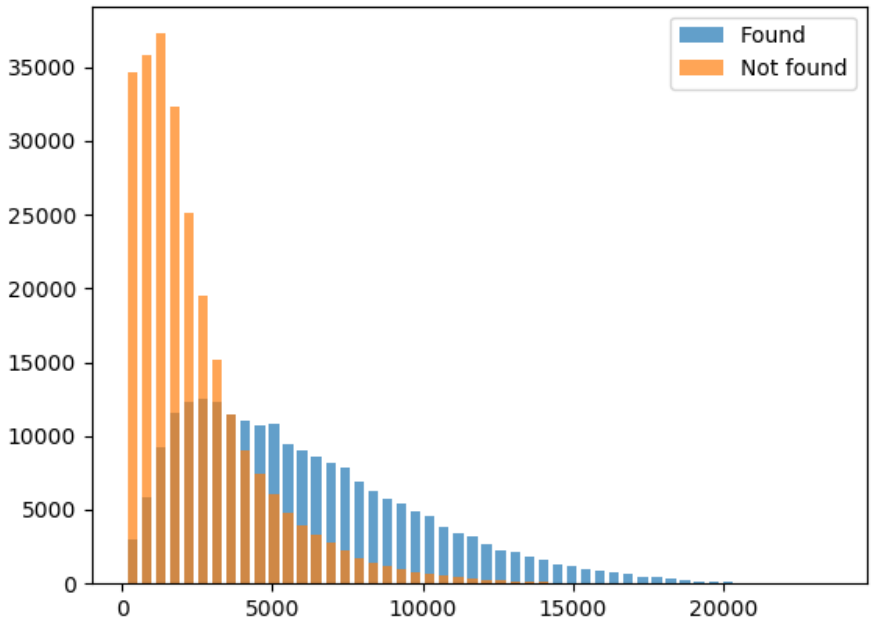

Figure 2: Overlap size distribution for k =10 with 1024 permutations and 0.04 threshold

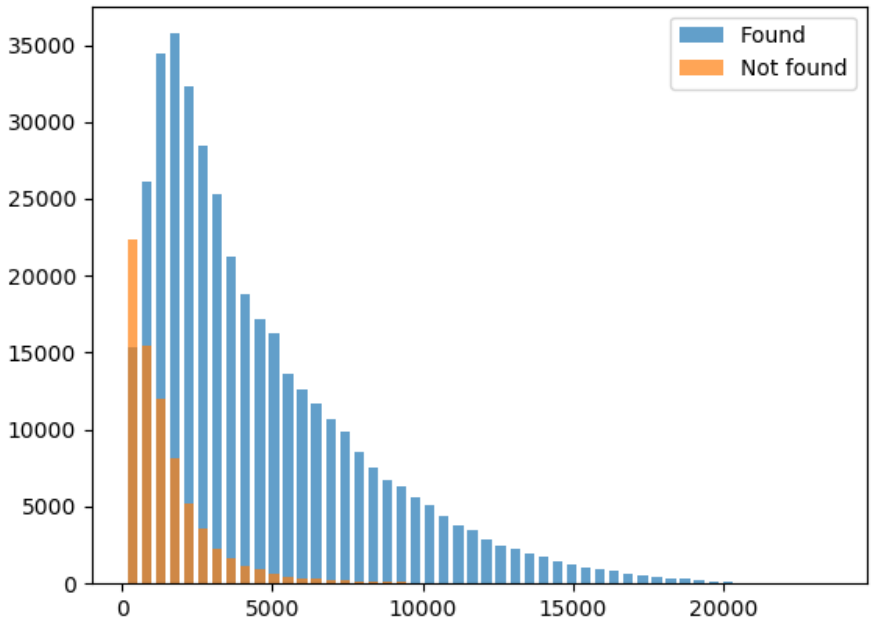

Figure 3: Overlap size distribution for k =12 with 1024 permutations and 0.01 threshold

Generally speaking, most overlap pairs we are missing have smaller overlap size compared to those we reported.

And when we decrease the threshold, we can find that more overlap pairs with smaller overlap size will be included in our final reported results.

2: simulated viral metagenomics

For this dataset, I followed the suggestions that if several reads are hashing to one bucket in LSH, then they are treated as one cluster. So the expected number of clusters is mainly determined by the choices of k, hashing function, and number of permutations(but seems this not really change anything in my experiment).

So I determine the properties of cluster by major vote from the reads in the cluster. If there are more reads from HCV, then I believe this cluster should belong to HCV, and all reads from HGV or HIV will be treated as error. By doing this I can made a confusion matrix for each experiment.

I also attached the cluster result for k = 10 in handin. Each line is a single cluster in the file.

(a)

I found the most important parameters is the size of kmer.

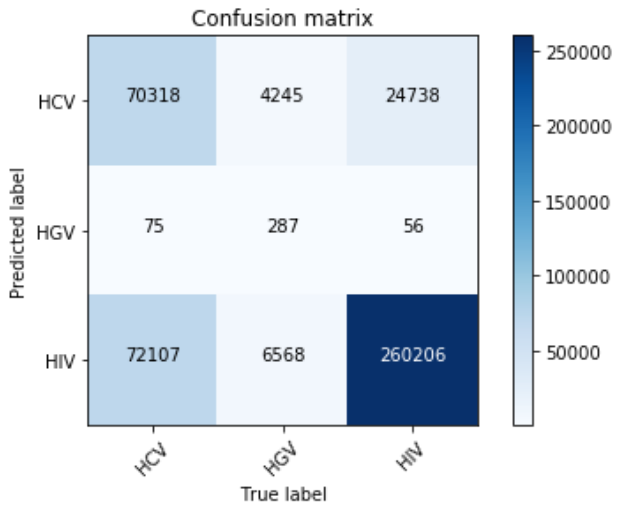

Figure 4: Confusion Matrix with k = 6

1. For k = 6:
   For 438601 reads we generate 1358 clusters
   70.81% HCV reads classfied into right clusters (HCV dominated cluster)
   68.66% HGV reads classfied into right clusters (HGV dominated cluster)
   76.78% HIV reads classfied into right clusters (HIV dominated cluster)

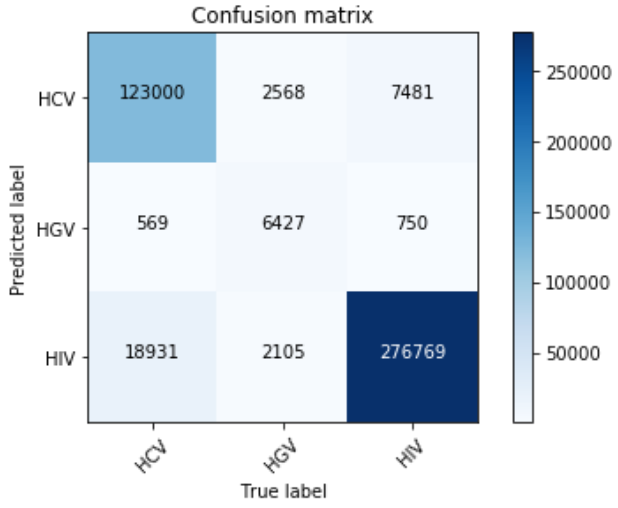

Figure 5: Confusion Matrix with k = 8

2. For k = 8:
   For 438601 reads we generate 9503 clusters
   92.45% HCV reads classfied into right clusters (HCV dominated cluster)
   82.97% HGV reads classfied into right clusters (HGV dominated cluster)
   92.94% HIV reads classfied into right clusters (HIV dominated cluster)

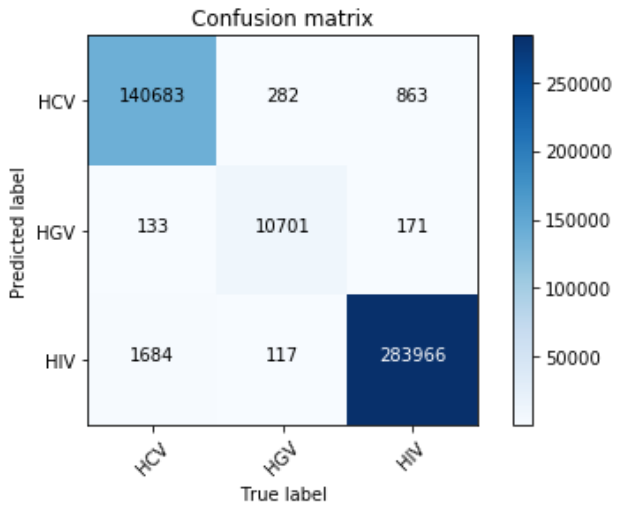

Figure 6: Confusion Matrix with k = 10

3. For k = 10:
   For 438601 reads we generate 21902 clusters
   99.19% HCV reads classfied into right clusters (HCV dominated cluster)
   97.24% HGV reads classfied into right clusters (HGV dominated cluster)
   99.37% HIV reads classfied into right clusters (HIV dominated cluster)

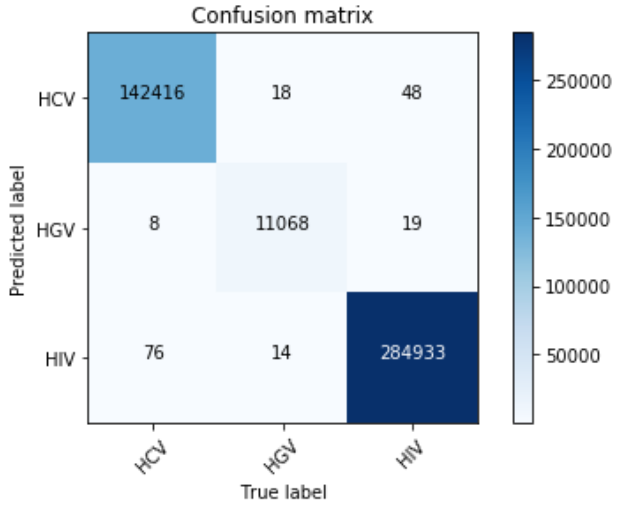

Figure 7: Confusion Matrix with k = 12

4. For k = 12:
   For 438601 reads we generate 26819 clusters
   99.95% HCV reads classfied into right clusters (HCV dominated cluster)
   99.76% HGV reads classfied into right clusters (HGV dominated cluster)
   99.97% HIV reads classfied into right clusters (HIV dominated cluster)

We can see, if k is too small, then our cluster cannot distinguish the difference of three viral, as

the kmer may not have enough variations. When we increase k, the classification error is smaller (as larger kmer can catch the signature information of that species) , but also we have much more clusters.

If our goal is to cluster all the reads into right clusters. Then it may be wise to choose larger k as beginning. Then later on we can merge small clusters to generate big clusters.