PlasmidHunter

Classifying plasmids and chromosomes in bacterial DNA

Hitesh Arora (11010122)
Rajarshi Chattopadhyay(11010274)

Vishal Vaibhav(11010270)
Krtin Kumar (11012119)

Abstract- In biological research, new generation sequencing data is increasingly used in different analysis such as drug-discovery, cancer research etc. In bacteria, plasmid is the circular DNA which carries antibiotic resistance, and plays a role in horizontal gene transfer. Hence it is an important problem to classify plasmid from chromosomes, but current sequencing methods are unable to efficiently do it. In this work, we use the approach of filtering using reference chromosomes, and apply machine learning methods of Hidden Markov Model (HMM), Support Vector Machine (SVM) and Neural Networks. We got an accuracy of 67.7%, 82% and 87.6% respectively in these methods, and analyzed their performance measures for different cases.

I. Introduction

A bacteria has a chromosomal DNA and one or more circular DNA molecule(s) called plasmids. Plasmids carry genes that may benefit survival of the organism (e.g. antibiotic resistance), and can frequently be transmitted from one bacterium to another (even of another species) via horizontal gene transfer. Hence, it is important to study plasmids in order to understand bacterial resistance and design effective antibodies. The current biological research is making exponential progress due to the availability of New Generation Sequencing (NGS) data. But the most popular sequencing technologies like Illumina generate short reads which makes it difficult to assemble plasmids, because of many repeat regions and mobile genetic elements in the plasmids. Hence it becomes a difficult and an important problem to identify plasmid sequences from chromosome sequences from the contigs (DNA sequence) obtained after sequencing a bacterial strain. Therefore, biologists have been working on this problem by trying various approaches, as it will greatly advance research in drug-discovery, understanding plasmid functions etc.

We begin with the approach of filtering using reference chromosomes, and then apply machine learning techniques of hidden markov model (HMM), neural networks and support vector machine (SVM).

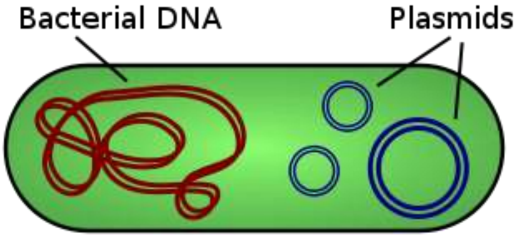
II. Problem

A. Problem Definition

Given all the contigs after sequencing a strain, we aim to build a tool to classify each of them as either a chromosome contig or a plasmid contig.

B. Data Description

The chromosome and plasmid data of E.Coli bacteria has been obtained from the Beatson Lab at The University of Queensland. The chromosomes have been obtained by the Illumina sequencing technology, and the plasmids from the newer PacBio sequencing technology which generates longer reads. This work will greatly help to utilize the large amount of Illumina data already available.



III. Filtering with chromosomes

There are certain differences (like anti-microbial resistance gene in plasmid), and certain similarities (insertion sequences) between plasmids and chromosomes. Using the fact that chromosomes and plasmids have (majorly) different sequences, we map (blast) all contigs of a given strain to a set of complete reference chromosomes allowing certain edit distance threshold. We call the set of contigs that did not map to any of the chromosome reference genomes as *Potential Plasmids*. Since the *potential plasmids* did not map to any of the reference chromosomes, so it is expected that plasmid contigs should be contained in them.

A. Method

We used an available tool called CONTIGuator to map given contigs to a reference chromosome. When contigs of a strain are mapped to a reference chromosome, a set of *mapped contigs* and a set of *unmapped contigs* are generated as output of CONTIGuator. We extract the set of contigs that did not map to any of the reference (chromosome) genomes by taking the intersection of all the *unmapped contigs* obtained by mapping the contigs of given strain to each reference chromosome.

We ran the experiment with different parameters to CONTIGuator (such as blastn vs mega blast algorithm, Min. Contig Length Threshold, Min. Contig Coverage Threshold) and obtained potential plasmids corresponding to each.
As per our discussion with researchers at Beatson Lab, we chose the parameters as mega blast algorithm; Min. Contig Length Threshold, L =200 bp; Min. Blast hit length as 200 bp; Min. Contig Coverage Threshold=20% and rest parameters as default for further analysis.

B. Testing with Controlled Data

NDM plasmids for some strains generated using PacBio sequencing technology were used for comparing with and analyzing Potential Plasmids. We did further work with those strains.
For each strain, we mapped the Potential Plasmid contigs of that strain to the assembled plasmid(s) of the corresponding strain. The mapping was done using CONTIGuator. If a strain had multiple plasmids then each of the plasmid was taken as reference one-by-one and PotentialPlasmids mapped onto it.

C. Results

We observed that in most cases potential plasmids mapped to large parts of the reference plasmids. The data is shown in the table below and a map for one strain is also shown.
Many of the potential plasmid contigs were not mapped to the reference plasmid, as observable from the table itself. The possible reason could be that those contigs correspond to the unique base sequences in the chromosome of the corresponding strain, which were not mapped to any chromosome in the given set of complete reference chromosomes.

Some of the regions of the reference plasmid were left unmapped too. This could be mainly due to two reasons-

- The unmapped reference region correspond to the contig in the plasmid which was also present in atleast one of the reference chromosomes and hence was mapped to it and not included in the set of Potential Plasmids. Such type of sequences are found in both plasmid and chromosome (like insertion sequences).

  This is one limitation of this approach, i.e. if some contigs belong to both chromosome and plasmid, they would not be included in Potential Plasmids, but rather marked as a chromosome contig. Therefore, we should appropriately choose the set of reference chromosomes to partition contigs effectively.

- The other reason could be that none of the contigs map to the unmapped regions (as shown in map) in the reference plasmid. This means that those reads/contigs weren't generated by Illumina sequencing.

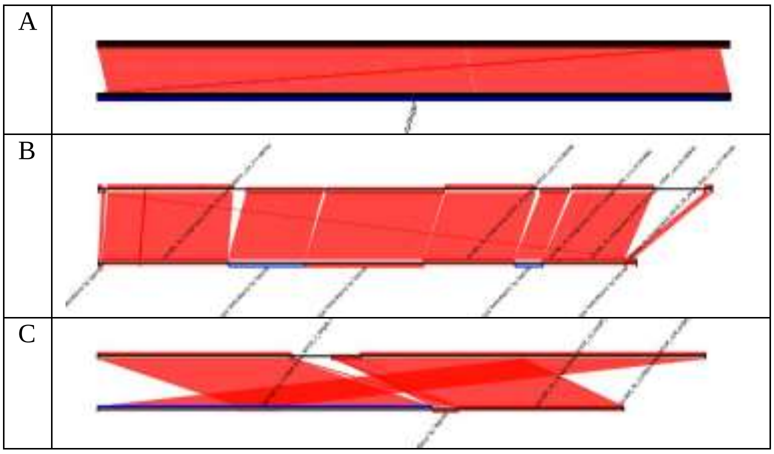
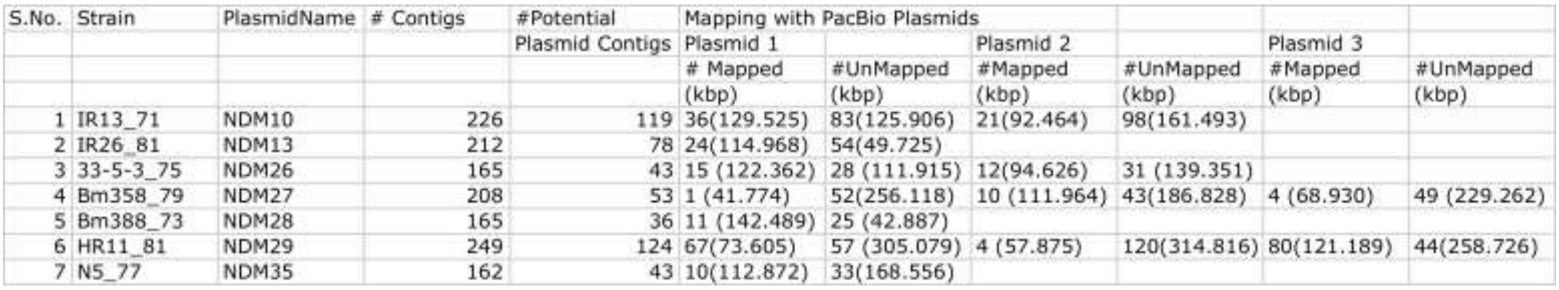


Figure: Mapping obtained using contiguator when potential plasmids of Bm358_79_Contigs was mapped against each of the plasmids (A) NDM27A, (B)NDM 27B, (C), NDM27C

D. Discussion

We have observed that this approach does find a substantial number of plasmids, but there are a lot of potential plasmids that are not mapped to actual plasmid. (False Positives).In order to reduce this set further, we plan to use the machine learning based methods to differentiate between plasmids and chromosomes.



S.No.	Strain	PlasmidName	# Contigs	#Potential Plasmid Contigs	Mapping with PacBio Plasmids					
					Plasmid 1		Plasmid 2		Plasmid 3	
					# Mapped (kbp)	#UnMapped (kbp)	#Mapped (kbp)	#UnMapped (kbp)	#Mapped (kbp)	#UnMapped (kbp)
1	IR13_71	NDM10	226	119	36(129.525)	83(125.906)	21(92.464)	98(161.493)		
2	IR26_81	NDM13	212	78	24(114.968)	54(49.725)				
3	33-5-3_75	NDM26	165	43	15 (122.362)	28 (111.915)	12(94.626)	31 (139.351)		
4	Bm358_79	NDM27	208	53	1 (41.774)	52(256.118)	10 (111.964)	43(186.828)	4 (68.930)	49 (229.262)
5	Bm388_73	NDM28	165	36	11 (142.489)	25 (42.887)				
6	HR11_81	NDM29	249	124	67(73.605)	57 (305.079)	4 (57.875)	120(314.816)	80(121.189)	44(258.726)
7	N5_77	NDM35	162	43	10(112.872)	33(168.556)				

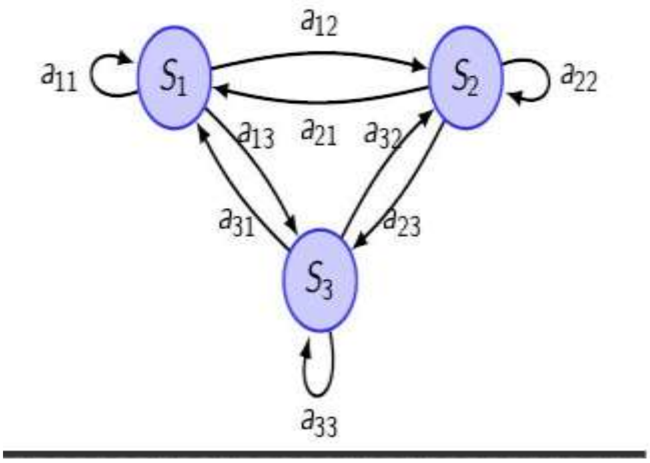
Results of mapping: #Contigs denote the number of contigs in that strain and #Potential Plasmid Contigs denote the number of contigs that were found as the Potential Plasmid contigs after running FindPotentialPlasmid.py. The Potential Plasmid contigs were mapped with actual plasmids (generated by PacBio) belonging to corresponding strains. The number of contigs mapped and the total number of bases (in Kilo base pairs) in those mapped contigs is shown in the parenthesis

IV. HIDDEN MARKOV MODELS

A Hidden Markov model (HMM) is a doubly embedded stochastic process, with an underlying stochastic process that is not observable (is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations.

A HMM contains several states a probability distribution corresponding to each state and transition probability between the states. Mathematically a HMM can be described by a set of five parameters λ = (A,B,π,N,M). 'A' is known as the transition matrix and contains the values of the transition probabilities between the states, 'B' is the matrix containing the value of the probability distribution of each state, 'π' is a vector containing the prior probabilities of each state, N denotes the total number of states and M denotes the number of symbols in each state. In practice HMM's are used in various applications, especially where analysis of temporal or spatial data is required.

Some example application areas include speech recognition, cursive handwriting recognition, stock market analysis, biological data analysis etc. HMM's also have an elaborate mathematical theory and which can be used to justify its use in these applications. The Forward-Backward Algorithm and the Viterbi Algorithm together provide a very efficient training and testing framework. The following figure shows a three state HMM:



Biological sequence like chromosomes contain a sequence of nucleotides. These sequences contain important information regarding generation of proteins, growth etc. Since the nucleotide sequences in some sense are analogous to speech signals, we thought that perhaps the algorithms that perform well in speech recognition might also perform well in recognizing nucleotide sequences. That was our motivation behind applying HMM to the problem of classifying contigs, as whether they come from chromosomes or from nucleotides.

The basic approach was to fit a three state fully connected HMM to each class of contigs and given an input contig, output whether it is from a chromosome or a plasmid using maximum likelihood principal. In our case the number of symbols per state is four. Mathematically if we denote the input contig using 'X' we can write the above classification rule as follows:

Output Class = Arg $_{\lambda}$ ***Max P(X| λ)***

Here 'λ' denotes the HMM models as stated earlier.

The following table shows the results and also presents some inference based on the results:

Results	Correctly Classified	Wrongly Classified
Chromosomes (Total 270 test contigs used)	183 (67.77%)	87 (32.22%)
Plasmids (Total 270 test contigs used)	148 (54.81%)	122 (45.19%)

Inference and Conclusion

The Average accuracy obtained was 61.29%. The HMM, with the sequence information directly as input did not perform very well on this classification task. Perhaps using some features would give better results. Following sections describe the use of classifiers like Support Vector Machines and Neural Networks. We have experimented with these classifiers using a particular type of feature.

V. SUPPORT VECTOR MACHINE

Support vector machine (SVM) is a popular supervised learning model used for classification of data. One of the attractive qualities of this classifier is its ability to perform non-linear classification by the use of kernels. Kernels project the data into a higher dimension space and as per Cover's Theorem the probability of being linearly separable increases on being projected to a higher dimensional space. In our experiment, we have chosen Gaussian kernel to map our features to higher dimensions.

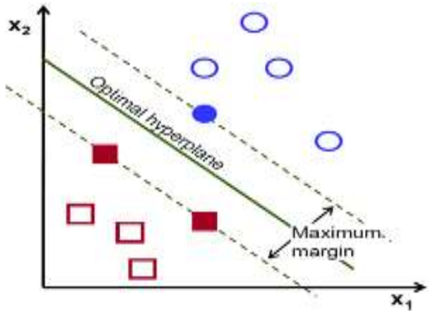


Fig. Representation of SVM as classifier

The major challenge while classifying plasmids and chromosomes is to extract features from the given genome sequence. Based on the importance of length 2 and length 3 genome sequence (encoding amino acid information) as information carrying sequences shown by independent biological research, we decided to take the count of all the length 2 and length 3 sequences present in the genome sequence and treated it as feature vector. In order to establish the significance of the length 2 and length 3 genome sequences, we experimented with several combinations of features taking length 2 and length 3 together and length 3 separately.

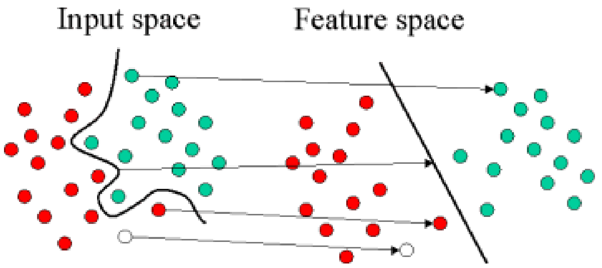


Fig. Representation of the importance of feature extraction and kernel.

	Case 1 :	Case 2	Case 3	Case 4
Training accuracy	83.96%	55%	83.96%	61.57%
Test case accuracy	81.86%	33%	82.13%	43.2%

Table showing results in several cases.

Case 1: Taking entire data into consideration

We take both length 2 and length 3 sequences into account while forming our feature vector and train our SVM. The results are shown in the table above. So the length of the feature vector was 80.

Case 2: Length normalization

Since we are breaking the plasmids and chromosomes into variable length sequences, we decide to make the feature vector independent of length of sequence by dividing it by the length of the corresponding sequence and then train the model. As obvious form the table, the approach doesn't lead to favorable results so we abandon this.

Case 3: Considering only length 3 sequences

In this case we consider only length 3 sequences to form the feature vector. As apparent form the above table showing results, we note that training accuracy remains same and test case accuracy increases slightly. The length of these feature vectors is 64.

Case 4: Considering only length 2 sequences

In this case we consider only length 2 sequences to form the feature vector. As the above table shows, the performance worsens in this case and hence length 2 sequences are not important in the classification. The length of the feature vector in this case was 16.

VI. Neural network

We have used a single hidden layer neural network to classify the contigs into either plasmids or chromosomes. A neural network depends upon various factors such as the Activation Function, Multi-Layer or Single-Layer, Forward or Backward and Number of Hidden Neurons.

A. Activation Function

The Activation Function is a fundamental concept in neural networks and is responsible for high flexibility. The Activation Function defines the function to fire a neuron depending on input pattern.

B. Multi Layer or Single Layer

Multi layer neural networks are networks that contains multiple layer of neurons, these layer receive input from other layers and output to other layers. All layers which are not input or output layer is known as hidden layer. Multi Layer Neural networks are very important for learning classification boundaries that is not linear in nature.

C. Forward or Backward

Each layer in a multilayer neural network can output either only in the forward direction or could serve as input to layers behind it, in this case it is known as Back Propagating Neural Network (or recurrent neural networks) and in the former case it is just a Forward Propagating Neural Network (or feed-forward neural networks)**.** Backward propagating neural networks are more powerful but also more resource intensive.

D. Number of Hidden Neurons

The hidden neuron can influence the error on the nodes to which their output is connected. The stability of neural network is estimated by error. The minimal error reflects better stability, and higher error reflects worst stability. Excessive hidden neurons will cause over fitting; that is, the neural network will overestimate the complexity of the target problem.

E. Output for Different Cases of Feature Vector

Case1

This feature vector gave a good accuracy of **84.6%.** 8.3% of the time the neural network falsely predicted a **Plasmid** and **8.1%** of the time it falsely predicted it to be a **Chromosome**. This was achieved with **11 hidden Neurons**, further increasing the hidden neurones led to over fitting of the training data and accuracy of test data reduced.

Case2

On the basis of weights of different features, it was found out that the features 'AGC' and 'CGA', were redundant as they neither affected positively or negatively to learning. Thus removing this led to an i**ncrease in accuracy to 85.5%** and reduction in **false alarm rate to 8.9% and 5.6%**. There was a slight increase in false alarm rate for plasmids suggesting that these features might be more important for plasmids.

Case3

A further increase in accuracy to 87.6% was seen when we considered only three length sequences as feature vector. This indicates that three length feature vector is more important than two length and pattern exists between 3 length and not 2.

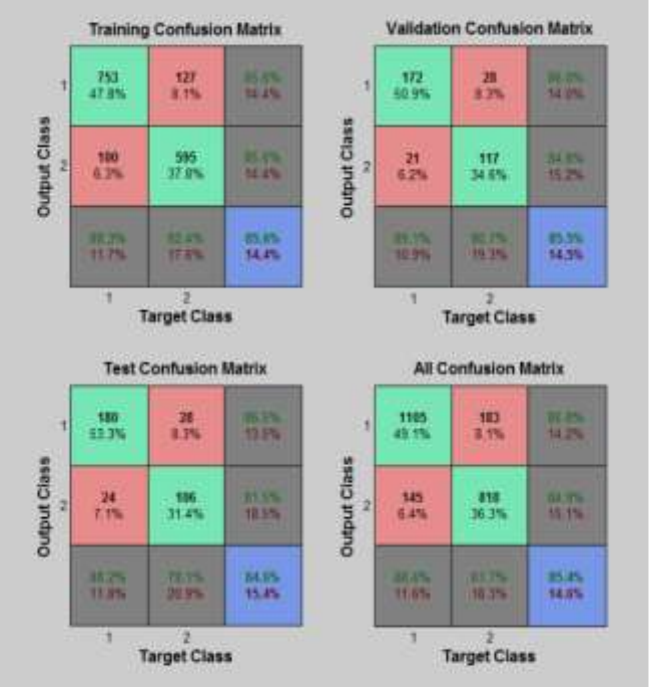


Fig- Case1: Features consists of 2 and 3 lengths of Genome Sequence

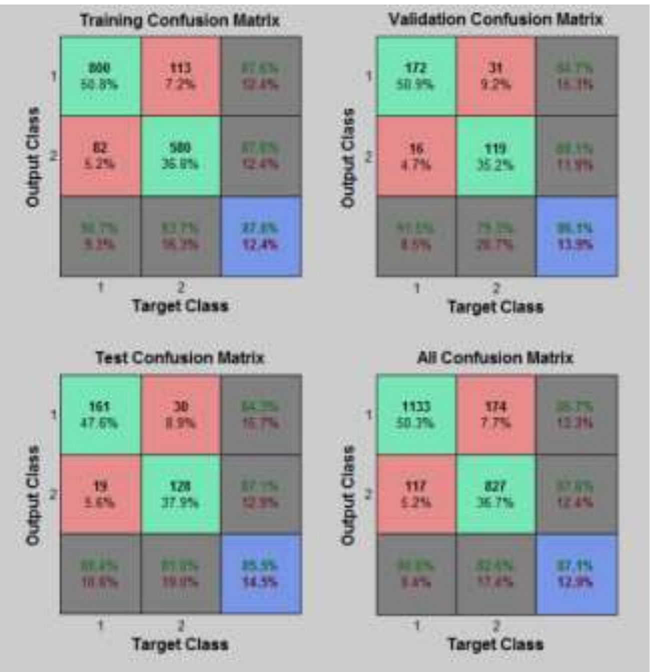


Fig- Case 2 Feature Vector consists of 2 and 3 lengths but reduction of ‘CGA’ and ‘AGC’

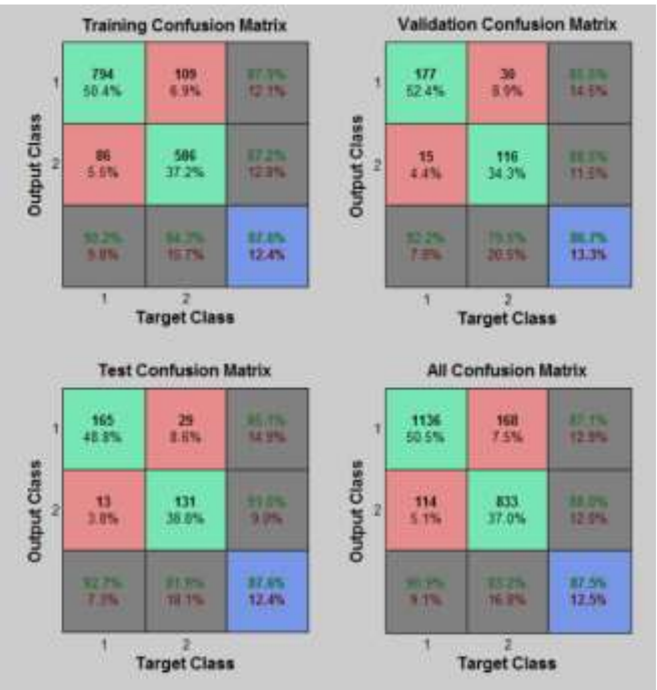


Fig- Case 3 Feature vector consists of only length 3 sequences

VII. Inference and Feature Modelling using HMM

As shown by the experiments earlier, using the HMM model and taking the nucleotide sequence directly as input to the model, the classification accuracy obtained was not good. However 67.77% accuracy obtained for chromosomes and 54.81% for plasmids show that there is some classification based information in the order of occurrence of nucleotides in chromosomes and plasmids.

In our experiments with Support Vector Machines and Neural Networks we used the count of all length three nucleotide units (after length normalization) as feature vector. The classification results were promising with SVM giving a net accuracy of 81.86% and neural networks giving an accuracy of 87.6%. This shows that the used feature vector is able to capture the information necessary for our classification task. However extracting this feature over the entire contig results in the loss of sequence information i.e. the feature only gives information about the occurrence ratio of each length 3 nucleotide unit but does not give any information about their order of occurrence.

However our experiment with HMM suggests that there is some classification related information in the order of occurrence of nucleotides. So a natural extension would be to compute these feature vectors taking small analysis windows on the nucleotide sequence. So each nucleotide sequence is converted to a 64 x T feature vector sequence where the value of T depends on the frame shift parameter and length of the sequence. The results showed a significant improvement. We varied the size of the analysis window and studied its effect on system performance.

Currently a 3-state fully connected HMM was used, with a single Gaussian modelling the probability distribution of the feature vectors in each state. Due to insufficient data we could not perform experiments on testing accuracy by increasing the number of states and number of mixtures per state. The table shows the results.

VIII. Conclusion and Future Work

Motivating results were obtained when HMM was used on the feature vector sequences. As stated earlier more experiments are certainly required to see if even better performance could be obtained using HMMs with more states and number of mixtures per state. However we need to use even larger datasets for this. Another direction would be to replace the Gaussian probability distribution function in each state with a neural network as this approach has given promising results for other sequence classification tasks like speech and handwriting recognition. In these areas neural networks are being able to better represent the probabilities distribution of feature vectors. Hence this approach must be definitely tried here.

Results – HMM with features

Analysis Window Size	400	300	200
Accuracy Obtained (fraction correctly classified)	Chromosome: 171/210 (81.43%) Plasmid : 169/216 (78.24%)	Chromosomes: 173/210 (82.38%) Plasmid : 171/216 (79.17%)	Chromosome : 177/210 (84.29%) Plasmid : 197/216 (91.20%)
Net Accuracy	**(79.81%)**	**(80.75%)**	**(87.79%)**

Acknowledgment

We are grateful to Dr. Scott Beatson and Dr. Mitchell Stanton Cook from the University of Queensland for their guidance. We also thank Dr. Rashmi Dutta for giving her opportunity to do this project and learn a lot.

References

[1] Lanza, Val F., et al. "Plasmid flux in Escherichia coli ST131 sublineages, analyzed by plasmid constellation network (PLACNET), a new method for plasmid reconstruction from whole genome sequences." *PLoS genetics* 10.12 (2014): e1004766 I.S. Jacobs and C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[2] Carattoli, Alessandra, et al. "In silico detection and typing of plasmids using PlasmidFinder and plasmid multilocus sequence typing." *Antimicrobial agents and chemotherapy* 58.7 (2014): 3895-3903.

[3] Galardini, Marco, et al. "CONTIGuator: a bacterial genomes finishing tool for structural insights on draft genomes." *Source code for biology and medicine*6.11 (2011).

[4] Rabiner, Lawrence, and Biing-Hwang Juang. "An introduction to hidden Markov models." *ASSP Magazine, IEEE* 3.1 (1986