UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

BACHELOR THESIS No. 6181

Using Reference Database for Plasmid Prediction

Sanja Deur

Zagreb, June 2019

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING
BACHELOR THESIS COMMITTEE

Zagreb, 13 March 2019

BACHELOR THESIS ASSIGNMENT No. 6181

Student: **Sanja Deur (0036498773)**
Study: Computing
Module: Computer Science

Title: **Using Reference Database for Plasmid Prediction**

Description:

Plasmids are mobile DNA fragments that can be transferred between bacteria. They are important because they carry genes responsible for antibiotic resistance and play a significant role in spreading and increasing antibiotic resistance. The aim of this thesis is to develop a method which will differentiate between plasmids and bacteria chromosomes. The method should be based on sequence similarity comparisons. Compare the results with the existing method based on the BLAST algorithm. Analyze the k-length subsequences (kmers) for known bacteria chromosomes and plasmids and visualize the results. For method comparison use cross-validation and ROC curve.
Programming code should be thoroughly commented and it should use one of standard coding styles. Complete application should be hosted on GitHub.

Issue date: 15 March 2019
Submission date: 14 June 2019

Mentor:

Full Professor Mile Šikić, PhD

Associate Professor Swaine Chen, PhD
(co-mentor)

Committee Chair:

Marko Čupić

Assistant Professor Marko Čupić, PhD

Committee Secretary:

Associate Professor Tomislav Hrkać, PhD

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA ZAVRŠNI RAD MODULA

Zagreb, 13. ožujka 2019.

ZAVRŠNI ZADATAK br. 6181

Pristupnik: **Sanja Deur (0036498773)**
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: **Korištenje referentne baze za predviđanje plazmida**

Opis zadatka:

Plazmidi su mobilni dijelovi DNA koji se prenose između bakterija. Važni su zbog toga što nose gene odgovorne za otpornost na antibiotike i tako doprinose širenju i povećanju otpornosti na antibiotike. Cilj rada je razviti metodu koja će razlikovati plazmide od bakterijskih kromosoma. Metoda se treba temeljiti na usporedbi sličnosti sljedova. Rezultate usporediti s postojećom metodom temeljenom na korištenju algoritma BLAST. Potrebno je napraviti i analizu podnizova duljine k poznatih bakterija i plazmida te vizualizirati rezultate. Za usporedbu metoda koristiti unakrsnu validaciju i ROC krivulju. Programski kod je potrebno komentirati i pri pisanju pratiti neki od standardnih stilova. Kompletnu aplikaciju postaviti na repozitorij Github.

Zadatak uručen pristupniku: 15. ožujka 2019.
Rok za predaju rada: 14. lipnja 2019.

Mentor:

Prof. dr. sc. Mile Šikić

Izv. prof. dr. sc. Swaine Chen (komentor)

Predsjednik odbora za završni rad modula:

Doc. dr. sc. Marko Čupić

Djelovođa:

Izv. prof. dr. sc. Tomislav Hrkać

Foremost, I am profoundly grateful to my mentors, Professor Mile Šikić and Professor Swaine Chen, for their guidance, patience and sharing their immense knowledge with me.

Further on, I would like to thank my family and friends, especially my parents, for their unconditional support throughout this entire process.

Table of Contents

1. Introduction

Bacterial cell, amongst other, consists of chromosome and zero, one or multiple plasmids [1]. Chromosome contains most of bacteria's genetic material, and plasmids carry few additional genes which help the bacteria to overcome stressful situations, especially antibiotic treatments. Plasmids have the ability of replication and to transfer easily to other bacterial species [2][3].

The ever-growing and widespread usage of antibiotics has triggered antimicrobial resistance. Nowadays, having the global trade and travel in mind, as well as bacterial ability to transfer from one host to another, antibiotic resistance and virulence plasmids which inhibit antibiotics and lead to novel and untreatable diseases are rapidly spreading [4][5]. This phenomenon has led to further studies investigating mobile genetic elements [6].

The main goal of these studies is automation of plasmid prediction and reconstruction methods. Commonly used approach is short-read whole-genome sequencing (WGS), which encounters difficulties working on plasmids with a high number of repeating sequences, resulting in fragmented assembly with numerous short contigs of unclear origin (chromosome or plasmid) [7]. Today, it is possible to automatically predict short plasmids, but the problems occur whilst trying to predict plasmids longer than 50 kbp, particularly the ones containing aforementioned repeating sequences [1].

The aim of this thesis is to adopt a known strategy used for metagenomic classification, i.e. matching to a database of known species, to classify plasmids. Specifically, we will use matching to a database of known plasmids, assume that new plasmids would look like those, and afterwards use that prior information for classification. This approach would improve with more information the same way metagenomic classification does, where bigger and more complete databases are progressively being built, so when a new sequence emerges, there exists a good chance of inferring that it came from another organism which has already been

sequenced. Described strategy has the benefit of potentially getting better as databases get better. However, it is known that a k-mer strategy actually performs worse when there is too much data, because the k-mers start to clash.

Furthermore, classification is carried out on plasmids and chromosomes, which are parted in k-length fragments in order to mimic incomplete assemblies. Classification is implemented using the existing BLAST algorithm on the one hand, and alternative method which differentiates between plasmids and chromosomes developed in this thesis on the other.

The main concern that this thesis addresses would be to inspect the quality of classification for unknown strains, i.e. whether the classification works in this case, and if the answer is affirmative, how accurate the plasmid and chromosome differentiation is.

2. Preliminaries

This chapter consists of biological background, crucial for better understanding of this thesis, as well as the brief introduction to plasmid prediction methods and overview of the most significant efforts in this field.

2.1. Chromosomes and Plasmids

The DNA of most bacteria is contained in bacterial chromosome, a single circular molecule, which can range in size from only 160 kbp to 12 200 kbp [8]. In addition to the chromosome, bacterial cell often possesses one or multiple small circular (in some cases linear) DNA molecules, called plasmids [3][9]. Whilst the chromosomes are big and contain all the vital genetic information, plasmids are relatively small (usually from 1 to 300 kbp) and carry only a few additional genes, as shown in Figure 2.1 [10].

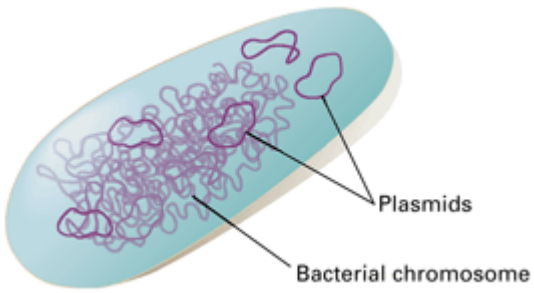

Figure 2.1: Bacterial chromosome and plasmids (size comparison) [11]

Plasmids are classified as mobile genetic elements (MGEs) due to their ability to self-replicate autonomously and transfer between bacteria (even of another species) [9]. This cell-to-cell transfer is called bacterial conjugation and it is a mechanism of horizontal gene transfer (HGT) [10].

Plasmids act as major drivers of bacterial evolution, environmental adaptation, genetic diversity and variation [5][7]. Moreover, they may carry genes which provide the possibility to destroy other bacteria, detoxification of harmful substances, and most importantly antibiotic resistance, which leads to the rise of multi-resistant pathogenic bacteria [2][3][6][12]. Latter represents major health care problem around the world, especially nowadays when people are accustomed to travel a lot, taking plasmid transportation trait into consideration [4].

Plasmids have been key to the development of molecular biotechnology and are also extensively used as tools in genetic engineering [3][5]. They are often used to introduce genetic material into bacteria, which leads to production of vital proteins (e.g. insulin, human growth hormone and so forth) [2].

To conclude, the importance of plasmids is unquestionable, and scientists are currently working on automating plasmid prediction and reconstruction methods, some of which are going to be presented in the Section 2.2.

2.2. Plasmid Prediction Methods

Exponentially increasing amounts of unprocessed bacterial genomic sequences are becoming available in public databases [13][14]. In the recent years, scientists have been making the efforts to analyse these sequences using automated methods. There are numerous extremely precise tools, but complete plasmid prediction from short-read sequencing data, as mentioned in Chapter 1, is not yet possible without taking certain manual steps [1].

In the following paragraphs, a short overview of the most important prediction methods is given, alongside with their advantages, disadvantages, and finally successfulness and attained results.

Some of the fully automated programs for plasmid prediction and reconstruction from whole genomes are: *PlasmidSPAdes, Recycler, cBar* and *PlasmidFinder.* An interesting study, referring to [1], has recently been conducted with its core goal being benchmarking four stated algorithms and determining the

possibility of obtaining complete plasmid sequences automatically. Programs are divided into two groups, *cBar* and *PlasmidFinder* which predict plasmids based on previously assembled contigs on the one hand, and *PlasmidSPAdes* and *Recycler* with their aim to reconstruct whole plasmid sequences on the other. Additionally, vastly successful method *PLACNET* would also belong to the latter group, but it will not be taken into consideration, because it depends on the expertise of the researcher, meaning it is partially manual.

Firstly, *cBar* predicts plasmids based on differences in k-mer composition, which is similar to method used in this thesis (see Chapter 4). Secondly, *PlasmidFinder* inspects replicon sequences and is mostly used for enterobacterial genomes. And finally, *PlasmidSPAdes* and *Recycler* search the de Brujin graph for plasmids.

The study concluded correct prediction for the vast majority of plasmids (89.9%), almost completely accurate for small and circular plasmids. It also confirmed the hypothesis of still challenging prediction of plasmids larger than 50 kbp.

Furthermore, there are methods working on a species level, such as *mlplasmids* and *PlaScope*. *On the one hand, mlplasmids* is using advanced machine learning algorithms, with support-vector machine (SVM) as a classifier to predict plasmid- and chromosome-derived sequences. For the purpose of training the model, *mlplasmids* uses short-read contigs from the following bacterial species: *Enterococcus faecium*, *Klebsiella pneumoniae* and *Escherichia coli*. *Mlpasmids* is shown to be superior to the previously mentioned algorithms, with its ability to accurately predict even large and linear plasmids. In contrast to previous methods, it is suitable only for genome assemblies from single species, but there is a possibility to train the model for more variety of species in the future [7]. On the other hand, *PlaScope* approaches the problem differently from all the above-mentioned methods, trying to combine both high sensitivity and specificity (for definition see formulae (4.1) and (4.2)). The method was tested on *Escherichia coli* and *Klebsiella* as well. It uses k-mer contigs, alike *cBar*, but is proven to be even more successful [14].

Some of the other mentionable methods are *PlasFlow, PlasmidTron, Plasmid ATLAS,* and *MOB-suite. PlasFlow* uses neural network approach to predict plasmidomes from environmental samples, thus having great impact on metagenomics. It is especially useful for analysis of large plasmids, and even linear ones do not represent an obstacle [15].

PlasmidTron is specific for utilizing phenotypic data from bacterial population studies to confirm presence of resistance genes in bacteria. It uses a k-mer based approach and filters out seldom k-mers. This tool was tested on *Salmonella enterica* and *Klebsiella pneumoniae* datasets [16].

Plasmid ATLAS is web-based tool with exceptional visual analytics tools used for analysis of high-throughput sequencing data. Its most important feature is quick identification of plasmids carrying specific antibiotic resistance genes [9].

Lastly, *MOB-suite* is able to identify contigs of plasmid origin with both high sensitivity and specificity. It succeeded in lowering the error rate, but that consequently led to splitting and merging of plasmid contigs. Another drawback is low accuracy prediction on novel plasmids, significantly differing from those already stored in the database [17].

The importance of this chapter lays in the fact that a lot of here described methods and datasets are used in this thesis as well. For instance, k-mer based approach, using k-length contigs in analysis and prediction. Furthermore, there will be efforts to classify some of the aforesaid bacterial species: *Escherichia coli, Salmonella enterica* and *Klebsiella pneumoniae.*

.

3. Data Summary

All bacterial genomes used in this thesis were retrieved from National Center for Biotechnology Information (NCBI) and are publicly available at ftp://ftp.ncbi.nlm.nih.gov/genomes. Only the genomes marked as 'Gammaproteobacteria' and 'Complete Genome' were taken into consideration. The total number of utilised fasta sequences is 7153, 1531 of which are chromosomes and remaining 5622 are plasmids.

4. Methods

This chapter presents methods used to classify, visualise and validate bacterial chromosome and plasmids. First, the classification method will be thoroughly described, both using existing BLAST algorithm, and an alternative algorithm developed in the thesis. Next, methods and options for plotting the different types of diagrams will be outlined. Finally, a brief description of cross-validation technique will be given.

4.1. Classification

Classification method attempts to classify incoming query, known or unknown, and determine whether it is a chromosome or a plasmid, based on the existing database containing both chromosomes and plasmids. There is also a third scenario when query is unclassified, i.e. falls into category ‘none’.

Even though, classification method can be used on whole chromosomes and plasmids, its most valuable usage is on fragmented chromosomes and plasmids. Fragmentation method is also developed and its aim is to fragment chromosomes and plasmids into k-length subsequences, in order to obtain set of simulated contigs.

Classification method is implemented both with BLAST algorithm and algorithm based on minimap2 developed in this thesis, both of which are going to be explicated in Chapter 5 and compared in Chapter 6.

Classification technique is carried out in three steps. First, query (chromosome or plasmid) is aligned with target (database), using either BLAST or minimap2 algorithm. First ten rows of BLAST output for plasmid *LN54850.1* are shown in Figure 4.1. Most important information in received output are name of contig which was aligned, subject name, i.e. target plasmid or chromosome which was hit, blast identity, i.e. percentage of successful mapping, and alignment length.

Next, self-hits, i.e. hits where query and target belong to the same chromosome or plasmid, are filtered out from the output. Method also contains optional tunable filter for blast identity, which will be referred to as chromosome or plasmid blast penalty cutoff further in the text. On the one hand, chromosome blast penalty cutoff is performed only on chromosomes. Chromosome is discarded if it has blast hit identity greater than cutoff, whereas plasmid blast hits with identity greater than cutoff are still kept. On the other hand, plasmid blast penalty cutoff is used only for plasmids, whilst it is ignored in the case of chromosomes. Plasmid is removed from further process if its blast hit identity is greater than cutoff. For example, looking at Figure 4.1, first seven rows will be omitted, because they represent self-hits. Furthermore, if plasmid blast penalty cutoff is set to 90%, ninth and tenth row will be filtered out because their identities are greater than cutoff. In this example, only ninth row will remain in output and be further analysed. Filtering results in classification being more difficult, with the purpose of making sure whether it is still reasonably accurate.

	Query	Contig	Subject	ID	AlignLen	Mismatch	Gaps	Qstart
1	LN554850.1	0_LN554850.1	LN554850.1	100.000	5000	0	0	1
2	LN554850.1	0_LN554850.1	LN554850.1	92.800	125	6	3	2489
3	LN554850.1	0_LN554850.1	LN554850.1	92.800	125	6	3	1724
4	LN554850.1	0_LN554850.1	LN554850.1	84.127	126	15	5	3000
5	LN554850.1	0_LN554850.1	LN554850.1	84.127	126	15	5	1727
6	LN554850.1	0_LN554850.1	LN554850.1	82.258	124	18	4	2999
7	LN554850.1	0_LN554850.1	LN554850.1	82.258	124	18	4	2491
8	LN554850.1	0_LN554850.1	LN554849.1	77.163	832	135	24	3708
9	LN554850.1	0_LN554850.1	LN554849.1	93.125	160	10	1	935
10	LN554850.1	0_LN554850.1	FM178383.1	90.816	196	18	0	3719

	Qend	Sstart	Send	Evalue	Bitscore	Qlen	Slen
1	5000	1	5000	0.00e+00	9234	5000	7177
2	2611	1724	1847	7.08e-41	178	5000	7177
3	1847	2489	2611	7.08e-41	178	5000	7177
4	3123	1727	1849	1.57e-22	117	5000	7177
5	1849	3000	3123	1.57e-22	117	5000	7177
6	3121	2491	2611	1.22e-18	104	5000	7177
7	2611	2999	3121	1.22e-18	104	5000	7177
8	4529	10537	9751	1.37e-117	433	5000	15266
9	1094	12589	12431	1.49e-57	233	5000	15266
10	3914	3318	3123	1.90e-66	263	5000	5360

Figure 4.1: BLAST output for plasmid *LN554850.1*

Finally, BLAST or minimap2 output data is summarized and outputted to a common file for multiple plasmids or chromosomes, which will be described in detail in Chapter 5.

4.2. Visualisation

Visualisation methods precisely represent parts of chromosome or plasmid query classified as chromosome, plasmid or none. Visualisation is of great importance whilst working with such abundant data sets as it is the case in this thesis.

A lot of different types of plots can be drawn using these methods, such as scatter plots, histograms, marginal histograms, pie charts, ROC curves and so forth. Implementation, available options and further description are given in Chapter 5.

4.3. Cross-Validation

Cross-validation is a model validation technique for assessing how the results of a statistical analysis (model) are going to generalize to an independent data set. It is mainly used in prediction problems when the aim is to estimate the accuracy of a predictive model when performed in practice [18].

The most important types of cross-validation are k-fold cross-validation and leave-one-out cross-validation [19]. The latter method will be used in this thesis in the sense of leaving the one species from the database out and afterwards trying to classify the exact same species.

The results of cross-validation procedure will be shown in the ROC (Receiver operating characteristic) probability curve which tells how much model is capable of distinguishing between classes, in our case plasmids and chromosomes. To plot the ROC curve, formulae for calculation of sensitivity and specificity must be introduced. Sensitivity (4.1) represents true positive rate (TPR) and is located on y-

axis, while specificity (4.2) represents false positive rate (FPR) and is situated on x-axis [20].

$$Sensitivity = \frac{True\ positive\ (contigs/bp)}{True\ positive\ (contigs/bp) + False\ negative\ (contigs/bp)} \quad (4.1)$$

$$Specificity = \frac{True\ negative\ (contigs/bp)}{True\ negative\ (contigs/bp) + False\ positive\ (contigs/bp)} \quad (4.2)$$

5. Implementation

This chapter deals with further explanation of methods defined in Chapter 4, brief overview of possible options whilst using these methods and other implementation aspects.

Programming languages used for methods' implementation, statistical analyses and visualising the results are Perl, R and Python. The most important scripts were posted on Github and are available at: https://github.com/lbcb-edu/BSc-thesis-18-19/tree/sdeur.

5.1. Classification

First of all, it is important to emphasise great significance of the following project: https://github.com/swainechen/closet on this thesis. Its code has been used to download and filter the database containing bacterial chromosome and plasmids described in Chapter 3. Parts of code were also utilised for different tests, visualisations and analytics. However, most importantly, some lines of code, which are essential for the thesis' implementation, are incorporated in *doBlast.R* and *doClassification.R* scripts.

Input of *doClassification.R* script are database (*Gamma_plasmids.fna)*, meta data about database content (*Gamma_plasmids_meta.txt),* directory containing chromosome or plasmid queries fragmented in k-length subsequences (in this project, specifically, k = 5000), and optional cutoff parameter previously defined in Chapter 4. Fragmentation is performed using *make-fragments.pl* script which comes, amongst others, with an option to choose the length of query subsequences. Another helpful script is *count-fragments.pl* which counts total number of fragments in aforesaid directory containing fragmented queries.

In *doClassification.R* script query sequences are compared with a database of sequences with either BLAST or minimap2 algorithm, which is described in detail

in Chapter 4. When trying to predict plasmids, output of the classification method would be length of well-classified part of contig (i.e. classified as plasmid), length of misclassified part (i.e. classified as chromosome) and remaining length which is considered unclassified (i.e. classified as none) and vice versa for chromosomes.

Part of the output for classification of plasmids from *Escherichia coli* is shown in Figure 5.1. Output consists of contig name, its total length and lengths of parts classified as plasmid, chromosome or none, only for chromosome hits greater than zero. Figure 5.1 indicates (e.g. line 7 – contig *9_AP017613.1*) that some of the plasmids are completely misclassified as chromosomes (100%), and some are still mostly classified as plasmids (e.g. line 2 – contig *2_AP009243.1*), even though there are also chromosome and none indices. This outputted file is of key importance in further analytics and visualisations of potentially ambiguous fragments.

Contig	Total	Plasmid	Chromosome	None
0_AP009243.1	5000	4707	293	0
2_AP009243.1	5000	4649	21	330
4_AP009243.1	5000	0	3722	1278
6_AP009379.1	5000	2441	2559	0
9_AP009379.1	5000	708	3372	920
16_AP010962.1	5000	4629	371	0
9_AP017613.1	5000	0	5000	0
0_AP018797.1	5000	3166	1834	0
1_AP018797.1	5000	1570	3430	0
2_AP018797.1	5000	4084	916	0
5_AP018797.1	5000	0	5000	0
5_AP018799.1	5000	0	5000	0
6_AP018799.1	5000	1051	3949	0
9_AP018800.1	5000	1926	3074	0
0_CP000799.1	5000	3458	1542	0
1_CP000799.1	5000	3838	1162	0
13_CP000799.1	5000	2555	2445	0
3_CP000799.1	5000	2996	2004	0
7_CP000799.1	5000	2201	2799	0
8_CP000799.1	5000	4932	68	0
9_CP000799.1	5000	2899	2101	0
1_CP002733.1	5000	1415	52	3533
11_CP002733.1	5000	4798	188	14

Figure 5.1: Output of the *doClassification.R* script for plasmids

5.1.1. BLAST algorithm

Basic Local Alignment Search Tool (BLAST) is an algorithm for comparing biological sequences, such as nucleotide or protein sequences to sequence databases, and calculation of the statistical significance of matches [21]. There are numerous types of BLAST algorithms for different purposes, with one of the most commonly used being nucleotide-nucleotide BLAST - *blastn* which will also be used in this thesis. This program, given a DNA query, returns the most similar DNA sequences from the DNA database [22].

doBlast.R script builds the BLAST database, conducts *blastn* command and afterwards filters out the self-hits and plasmids or chromosomes with identity higher than cutoff parameter (i.e. higher than 99%). This script is used in *myClassification.R,* but there is also a possibility to run it from command line.

Additional *blastn* options used in this implementation are:

- -out - -evalue 1e-10
- -dust no
- -outfmt "6 qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore qlen slen"
- -max_target_seqs 100

5.1.2. Minimap2 algorithm

Minimap2 is a versatile general-purpose pairwise alignment program to map DNA or long mRNA sequences against a large reference database. It is 3–4 times faster than mainstream short-read mappers at comparable accuracy [23].

The implementation is written in Python and R programming languages. First, minimizer index for the reference was created and saved with option -d, because that process can be time-consuming if done in a loop, which is the exact implementation of *doClassification.R* which utilizes this implementation.

Next, alignment of query chromosomes and plasmids is performed, using the following options: -N 100 and --secondary=yes. Option -N is used to define the maximum number of secondary alignments, i.e. not only the best hits, whilst option --secondary is used in order to output those secondary alignments. Minimap2's option -N is similar as BLAST's option -max_target_seqs.

Finally, the output is parsed using the Python script to obtain only the valuable information for classification method. Afterwards the *doMinimap.R* script filters the self-hits and hits with identity above cutoff value out, similar as the *doBlast.R* script. The *doMinimap* function is then called in *doClassification.R* script and output file is generated in the already described form (Figure 5.1).

5.2. Visualisation

Abundant data sets, alike those in the thesis, require excessive visualisation methods. They are implemented in R programming language and mostly concentrated in the following R scripts: *doPlot.R* and *statistics.R.* The *doPlot.R* script is automatically called in *plot-graph.pl* script which performs comparison between one or more queries and target database using the BLAST algorithm and at last commences the plotting of chromosome or plasmid hits on the chosen type of density plot.

There are a great deal of options in *plot-graph.pl* script as follows:

- amount: choosing between single or multiple queries
- type: choosing between plasmid or chromosome
- query: assigning of .fna file which will be used as query (available only in single query mode)
- hits: choosing the number of hits in blast method (available only in multiple query mode)
- cutoff: filtering out the hits with blast identity greater than cutoff parameter (i.e. greater than 99%)
- graph: plotting scatter plot or scatter plot with marginal histograms

Density diagrams show relationship between hit length, i.e. length of the aligned part of query to target, located on y-axis, and identity, i.e. percentage of successful mapping, situated on x-axis. In Figure 5.2 graph with default options can be seen, while Figure 5.3 demonstrates diagram with default options, except the 'graph' option which is set to plot scatter plot with marginal histograms. Furthermore, plotting of multiple chromosomes and plasmids is shown in Figure 5.4 and Figure 5.5, respectively.

In conclusion, most of the hits are correct, but some chromosomes hit to plasmids, and vice versa, which is the exact problem which will further be analysed in Chapter 6. Moreover, incorrect hits are usually those of short hit lengths which is in accordance with the thesis' introduction (Chapter 1) statement about short sequences.

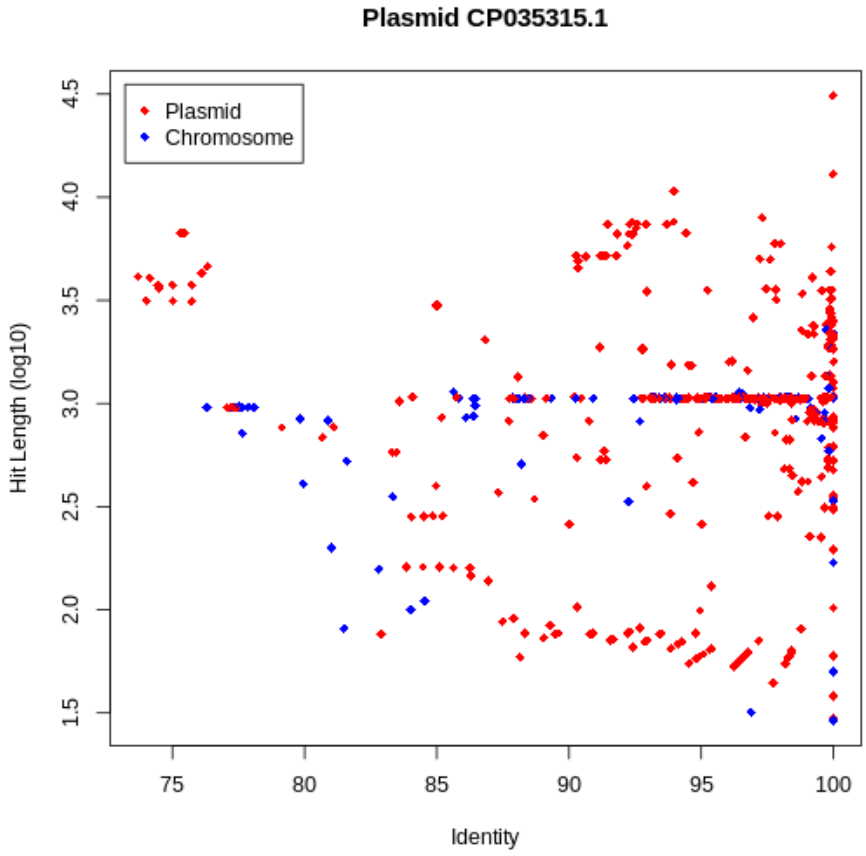

Figure 5.2: Density plot with default options

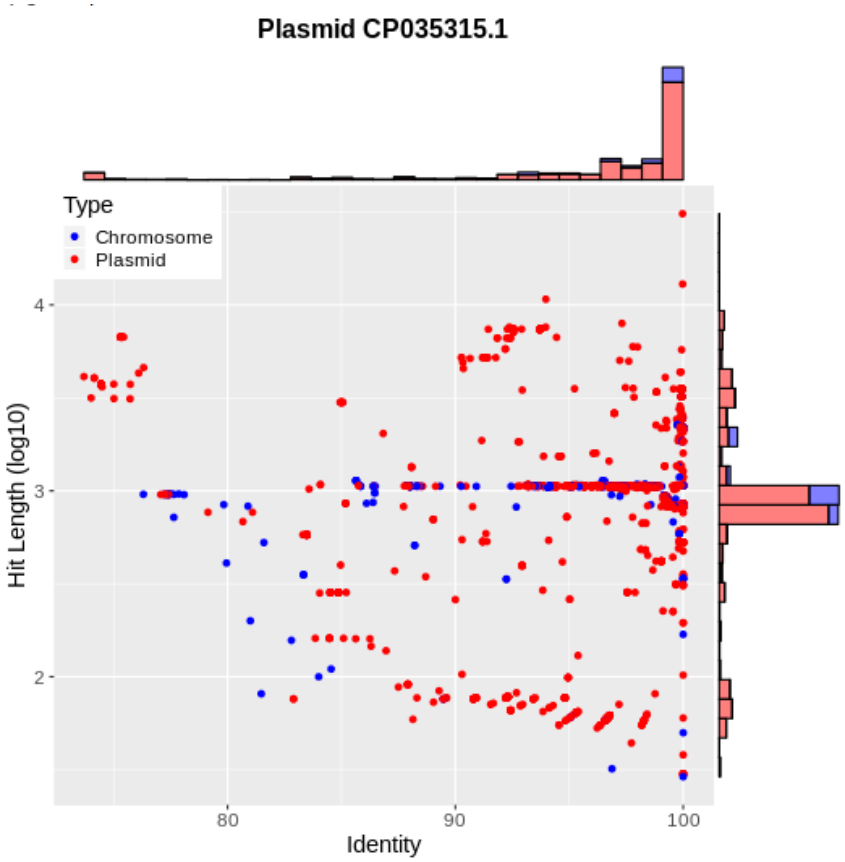

Figure 5.3: Density plot with default options and additionally marginal histograms

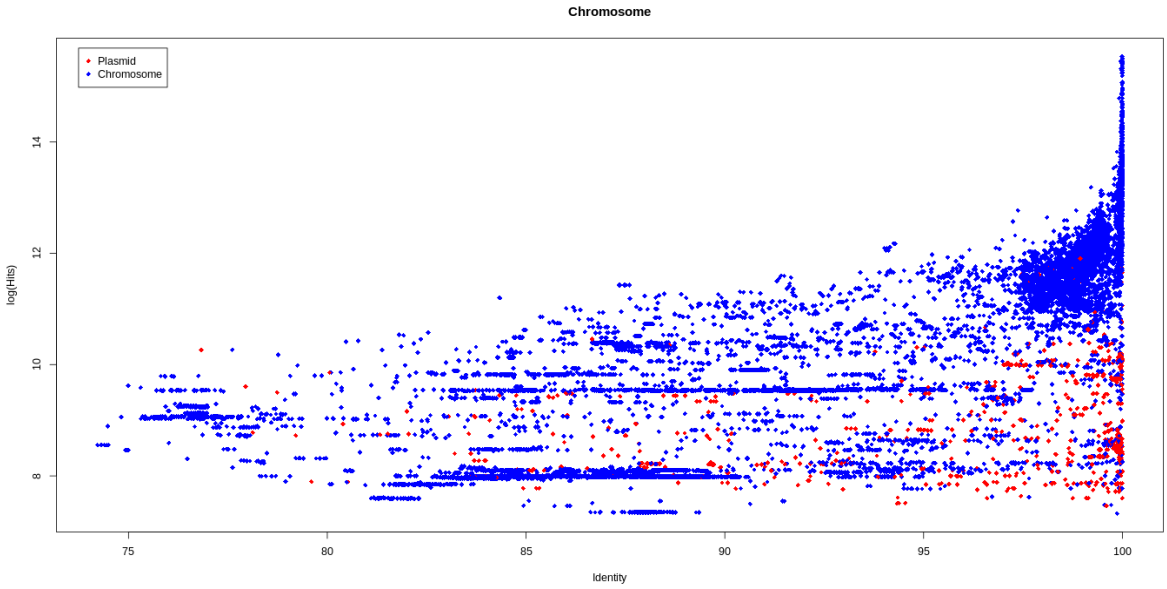

Figure 5.4: Density plot for multiple chromosomes (100)

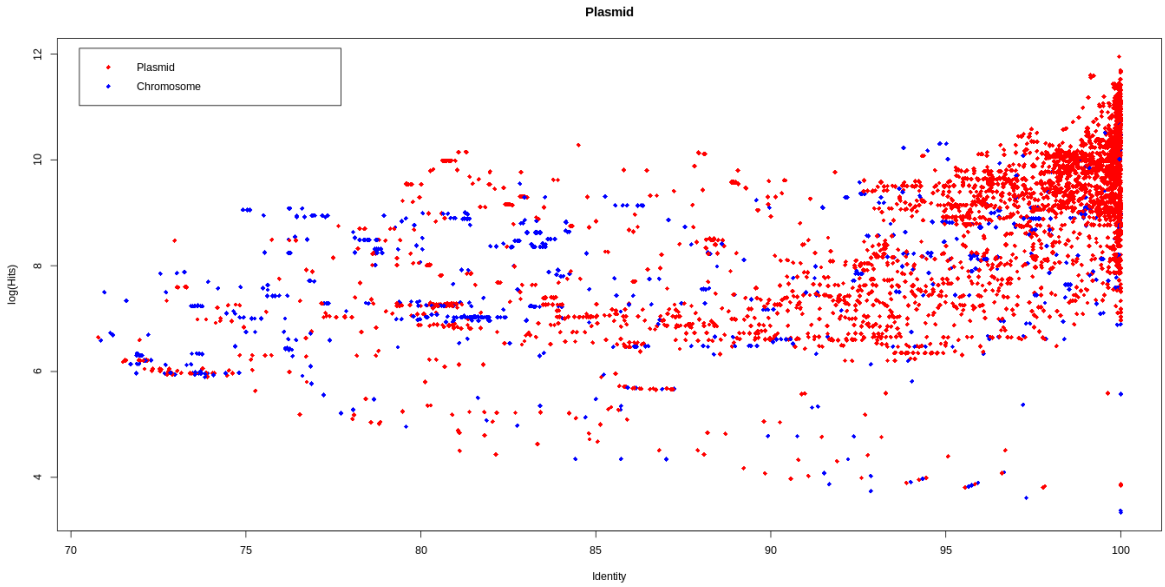

Figure 5.5: Density plot for multiple plasmids (100)

statistics.R has multiple usages in classification and cross-validation of chromosomes and plasmids. Its graphs display percentage of well-classified, potentially ambiguous and misclassified chromosome and plasmid fragments, as well as the resulting ROC curves. The detailed description and illustrations of the graphs will be given in the Chapter 6.

5.3. Cross-Validation

Cross-validation technique will be applied to the case of three bacterial species: *Escherichia coli* (1912), *Klebsiella pneumoniae* (1521) and *Salmonella enterica* (749), sorted in a descending order by number of their occurrences in the database. Besides the fact that these species are the most represented ones in the database, they are also of a great clinical importance, because of their pathogenic traits and ever-growing multi-drug resistance (MDR).

Implementation of cross-validation is conducted by leaving the species which are going to be validated, along with its closely related species out of the database. First, in the case of *Escherichia coli*, *Escherichia albertii* (28),

Escherichia fergusonii (7)*, Escherichia marmotae* (3), *Shigella boydii* (27)*, Shigella dysenteriae* (56)*, Shigella flexneri* (75) and *Shigella sonnei* (68) are left out. Next, alongside *Klebsiella pneumoniae, Klebsiella aerogenes* (63), *Klebsiella michiganensis* (27)*, Klebsiella oxytoca* (75)*, Klebsiella quasipneumoniae* (54)*, Klebsiella quasivariicola* (4)*, Klebsiella sp.* (22) and *Klebsiella variicola* (55) are omitted. Finally, beside *Salmonella enterica, Salmonella bongori* (2) and *Salmonella sp.* (15) are left out. Bacterial species are written alphabetically, alongside number of their occurrences in the database.

Results and corresponding visualisations for the described procedure will be given in Chapter 6.

6. Results and Discussion

This chapter will be divided into two crucial analyses – classification of fragmented chromosomes and plasmids, and leave-one-out cross-validation performed on three significant aforementioned bacterial species.

6.1. Classification

Classification procedure attempts to deliver correct classification of chromosomes and plasmids, or at least one with the least amount of inaccuracies. First, classification for randomly chosen single chromosome and plasmid is provided. Next, classification is conducted on whole dataset of plasmids (5622) and hundred randomly chosen chromosomes. The reason for choosing only hundred chromosomes is their size which makes the classification exceedingly time-consuming. Finally, there is an example of filtering out the best hit in every step, with the purpose of making the classification increasingly difficult.

Chromosome *AL513382.1* is randomly chosen, fragmented and classified at chromosome blast penalty cutoff of 100%, i.e. no cutoff, and cutoff of 99%. Figure 6.1 and Figure 6.2 illustrate the worsening of classification, but it is still considered acceptably accurate. Moreover, one of the contigs (*206_ AL513382.1*) with high percentage of plasmid hits is chosen and compared before and after performing the chromosome blast penalty cutoff and presumably its classification is much worse, which is shown in Figure 6.3 and Figure 6.4.

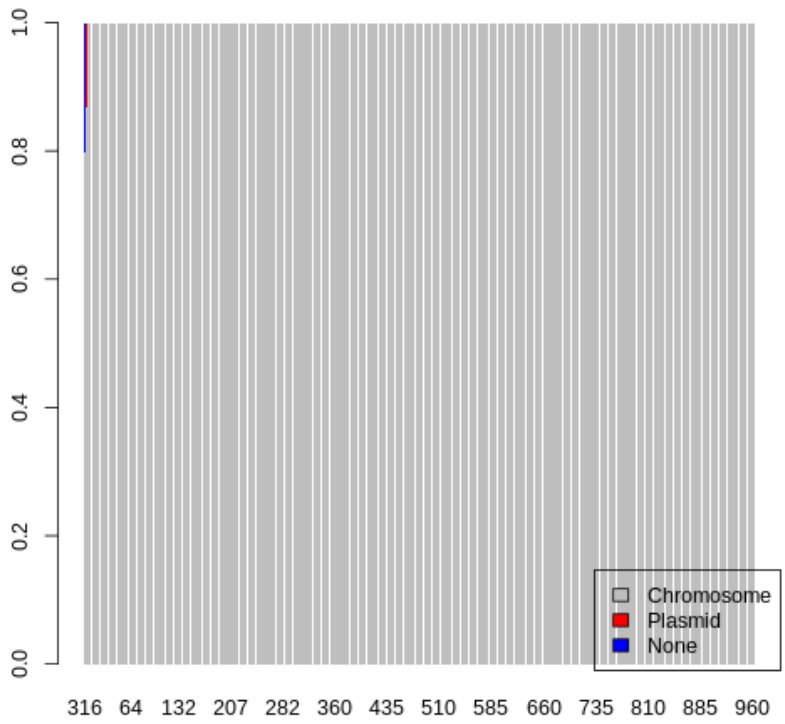

Figure 6.1: Classification of chromosome *AL513382.1* with no cutoff

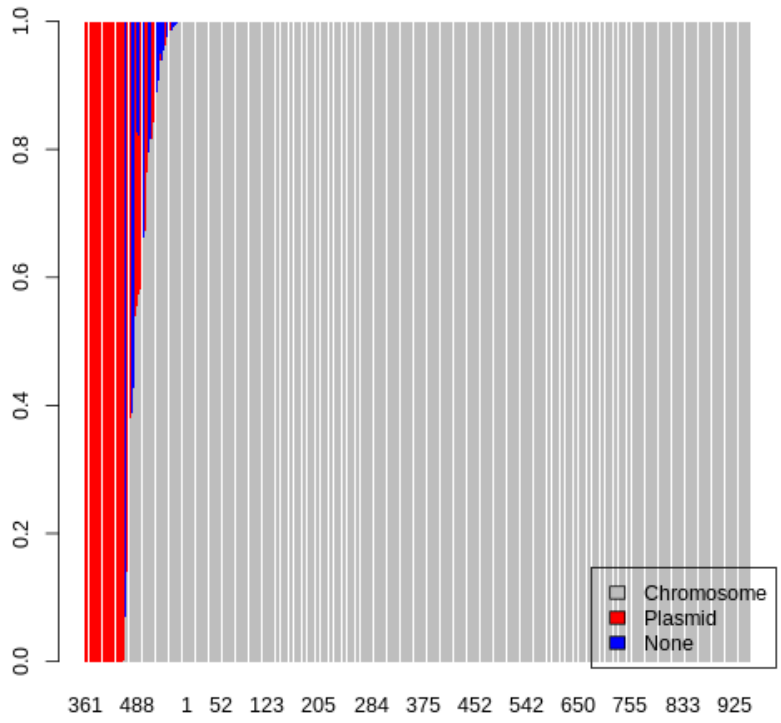

Figure 6.2: Classification of chromosome *AL513382.1* with 99% cutoff

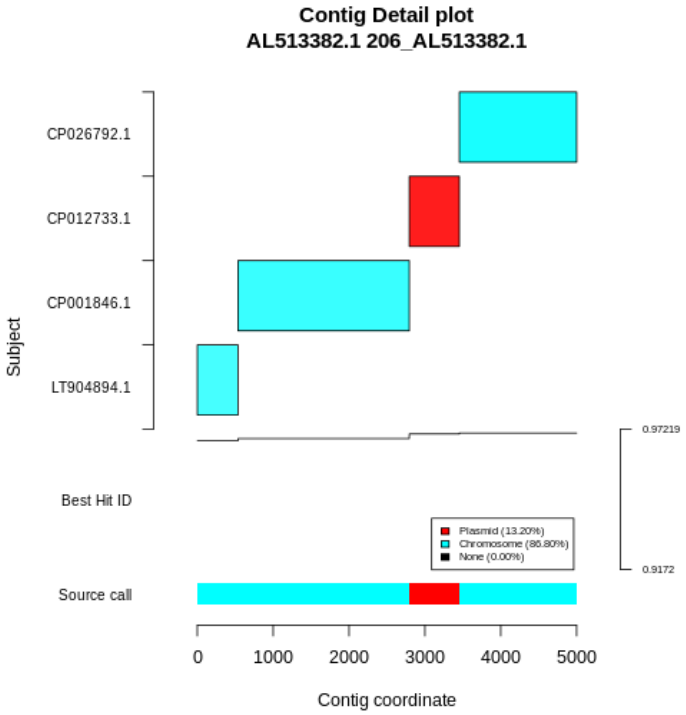

Figure 6.3: Detail plot of contig *206_AL513382.1* with no cutoff

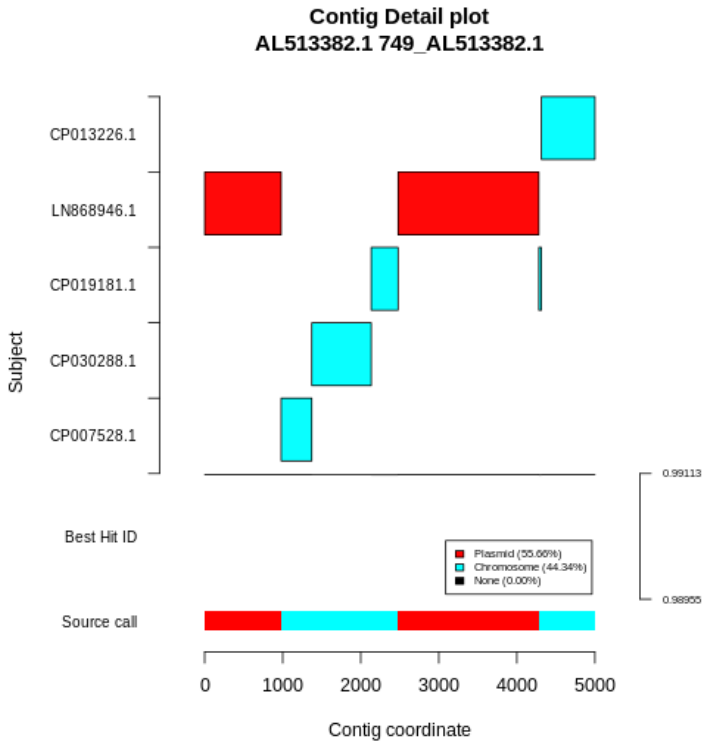

Figure 6.4: Detail plot of contig *206_AL513382.1* with 99% cutoff

Looking at Figure 6.5 and Figure 6.6, it can be concluded that classification of plasmid (*AL513382.1*) is worse, comparing to the one of chromosome in Figure 6.1. and Figure 6.2. Furthermore, the impact of plasmid blast penalty cutoff can be seen in Figure 6.8 which contains noticeably fewer plasmid hits than the case without cutoff illustrated in Figure 6.7.

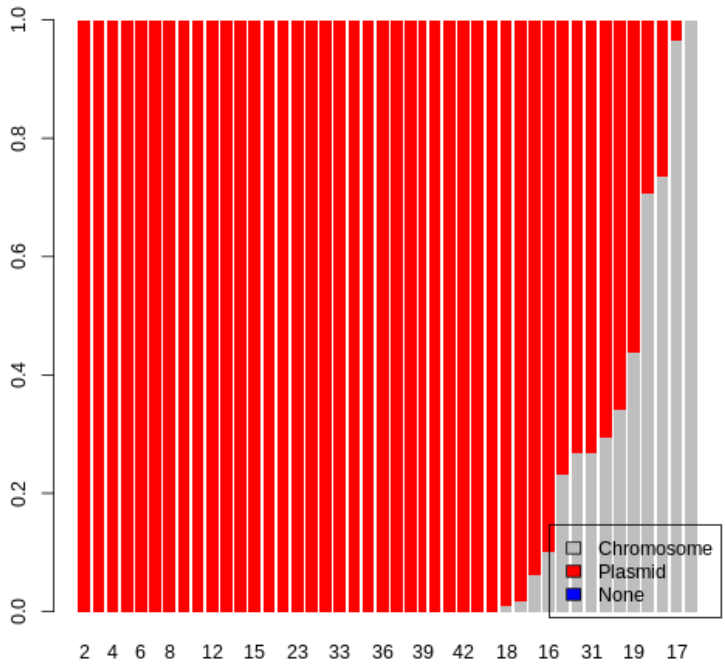

Figure 6.5: Classification of plasmid *AL513383.1* with no cutoff

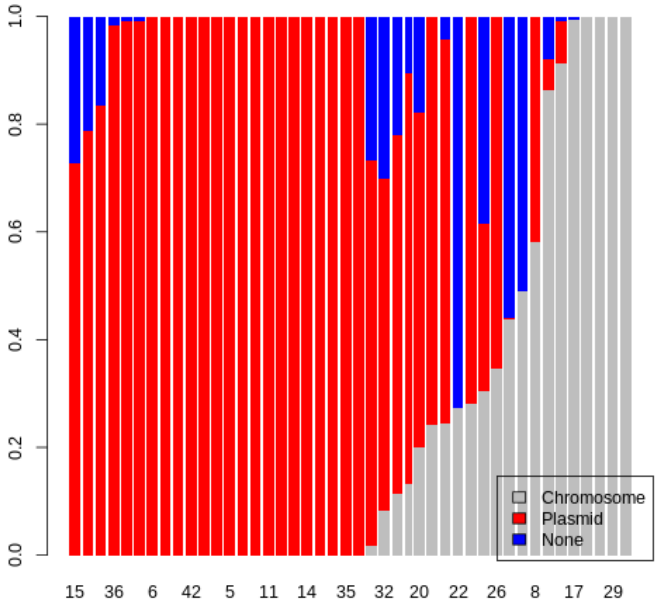

Figure 6.6: Classification of plasmid *AL513383.1* with 99% cutoff

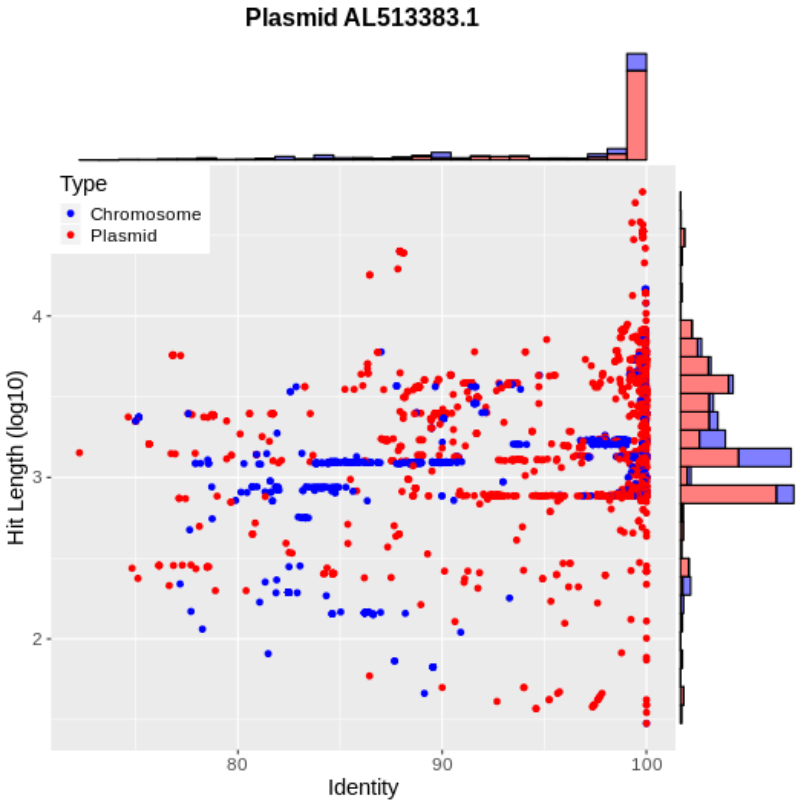

Figure 6.7: Density plot of plasmid AL513383.1 with no cutoff

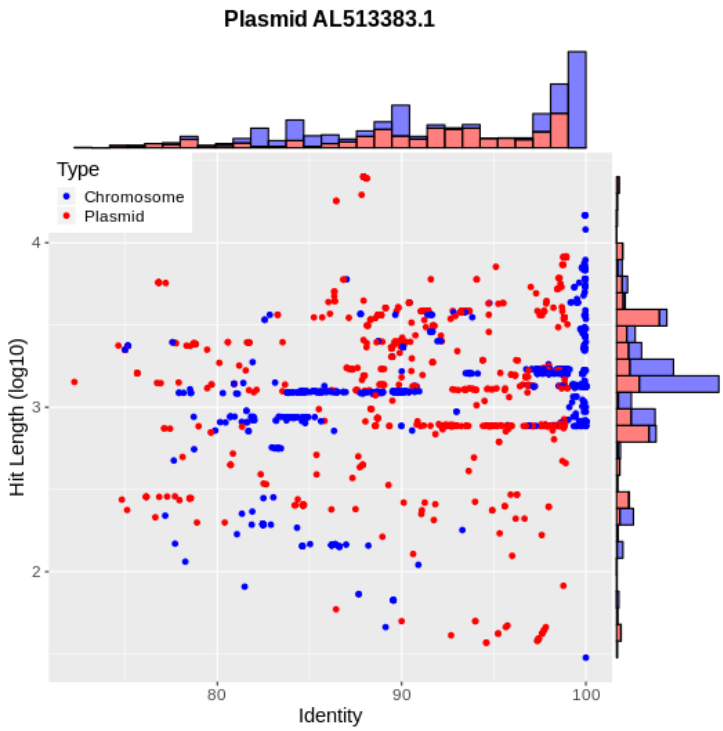

Figure 6.8: Density plot of plasmid *AL513383.1* with 99% cutoff

Above-mentioned conclusions were made only on the basis of single, randomly chosen chromosome and plasmid. In order to acquire outcomes of greater significance, classification is made for all plasmids from database and hundred randomly chosen chromosomes as described before. The test is again divided into two cases: no cutoff and cutoff of 99%. Outcome is shown in Table 6.1, where columns represent number of chromosome fragments which have >x% plasmid portion, divided by total number of chromosome fragments (6.1), and vice versa for plasmids (6.2).

In conclusion, remarkably similar results repeat for multiple chromosomes and plasmids, as they were for single ones, so generalisation is feasible and it is justified to take only hundred chromosomes into consideration. Moreover, chromosomes achieve better classification results than plasmids and introduction of cutoff worsens the classification results, which was also the case for single chromosomes and plasmids. Additionally, whilst examining the rows of Table 6.1, it can be seen that values quickly tend to drop to 0%. This fact will be well-illustrated in Figure 6.19 and Figure 6.20 in Section 6.2.

$$C = \frac{no.of\ chr.fragments\ with > x\%\ plasmid\ portion}{total\ number\ of\ chr.fragments} \times 100\% \qquad (6.1)$$

$$P = \frac{no.of\ pla.fragments\ with > x\%\ chromosome\ portion}{total\ number\ of\ pla.fragments} \times 100\% \qquad (6.2)$$

Table 6.1: Potentially ambiguous chromosome/plasmid fragments

	> 0 % plasmid/chromo	**> 25 % plasmid/chromo**	**> 50 % plasmid/chromo**	**> 75 % plasmid/chromo**
Chromosome, no cutoff	1.98 %	0.77 %	0.54 %	0.42 %
Chromosome, cutoff 99%	7.53 %	5.02%	3.79 %	3.24 %
Plasmid, no cutoff	24.32 %	12.13 %	6.10 %	2.34 %
Plasmid, cutoff 99%	27.17%	18.23 %	11.10 %	4.98 %

Third approach is removal of best hit and then doing the classification, in every iteration. First and second approaches eliminate only self-hits in the case of no cutoff, and hits with identity greater than cutoff for chosen chromosome or plasmid blast penalty cutoff. Method described in this section is depicted as a kind of a ‘middle-ground’ between filtering out only the self-hit, and possibly filtering out too many hits, using the cutoff parameter. Just filtering out the self-hits might not truly eliminate all self-hits, because there might be a very nearly identical plasmid in the database. For that exact reason, the results obtained by this method should be more accurate.

The flow of described experiment is shown in Figure 6.9, where it can be noticed that percentage of chromosome hits and ‘none’ (unclassified parts) is slowly rising. Grey part represents chromosome, red part plasmid and blue part none, i.e. unclassified portion. With every step forward, the graphs look more and more like the one in Figure 6.6. Furthermore, slow increase in chromosome and none sections is shown in Figure 6.10.

Lastly, it can be concluded that with the reduction of cutoff, i.e. identity filter for plasmid, next best hit is to a chromosome. This shows that classification technique rather misclassifies, than un-classifies, i.e. classify as none. The same conclusion can also be drawn for chromosomes as seen in Figure 6.2 – greater amount of plasmid, than none portion is present.

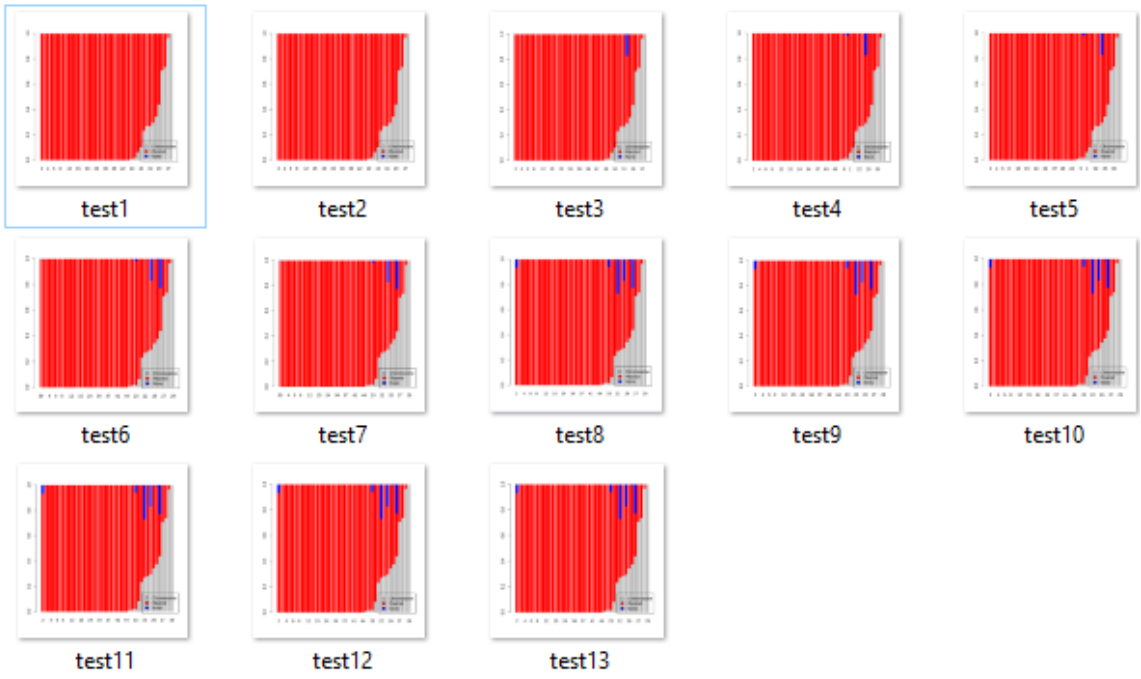

Figure 6.9: Flow of best hit removal experiment

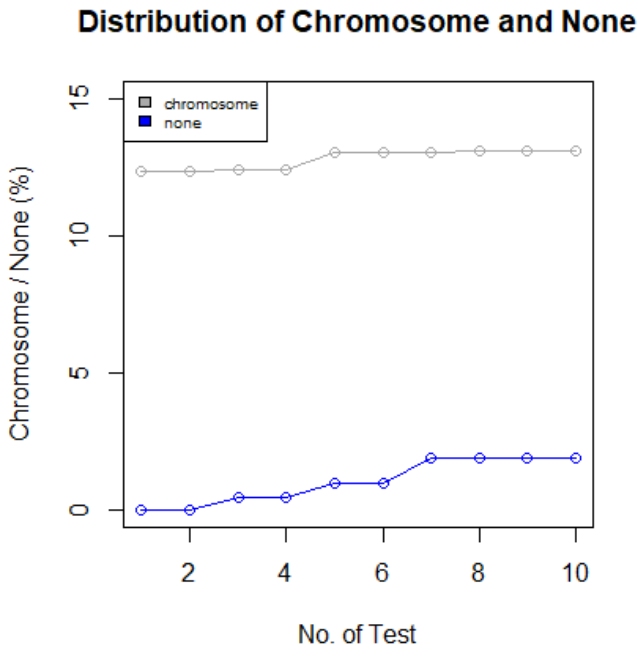

Figure 6.10: Increase of chromosome and none portion in the best hit removal experiment

6.2. Cross-Validation

Cross-validation is conducted on three fore-mentioned bacterial species, with *Escherichia coli* being the most frequent one in the database. There are eleven graphical illustrations for each species, but entirety of them will be shown only for *Escherichia Coli*, whilst other data will be stored in table form for easier analysis purposes. Diagrams for other two species can be omitted, because of their similarity to the ones for *E. coli.* ROC curves, as the most insightful visualisation methods in this case, will be shown for all three species. It is important to mention that all of the following graphs and calculations are made in *statistics.R* script.

First, there are four graphs for chromosome fragments and four for plasmid fragments, with no cutoff. In Figure 6.11 and Figure 6.15 relationship between correctly classified fragments (100% chromosome or plasmid, respectively) and potentially ambiguous fragments (>0% plasmid or chromosome portions, respectively) is shown. In Figure 6.12 and Figure 6.16 the portion of >50% plasmid or chromosome in above-defined potentially ambiguous fragments can be seen.

The reason for plotting the diagrams for specifically 50% is the possibility to classify chromosome as chromosome, and plasmid as plasmid fragment if the hit is >50% chromosome and plasmid, respectively. Figure 6.13 and Figure 6.17 show percentage of plasmid portions in potentially ambiguous fragments, whilst Figure 6.14 and Figure 6.18 show the same for 'none'.

Even though percentage of >0% plasmid is much higher than the one in the first row of Table 6.1, it can be seen in Figure 6.12 and Figure 6.13 that vast majority of them are still below 50%, which indicates that classification is still quite reasonable. Furthermore, results for plasmids which can be seen in the third row of Table 6.1 and Figure 6.15 are extremely similar, which is interesting considering that cross-validation was conducted for 'unknown, never-seen' plasmids, i.e. the ones which are not contained in the database. Moreover, in Figure 6.16 and Figure 6.17 it is illustrated that even though some fragments have chromosome portions, they are mostly quite low.

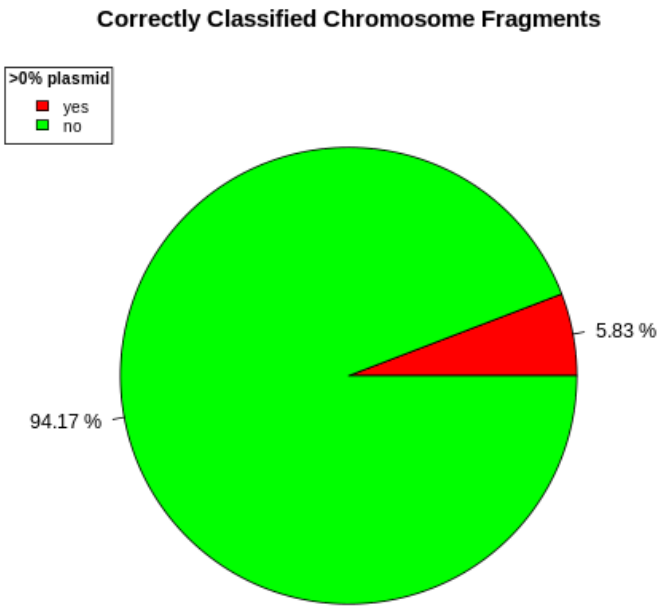

Figure 6.11: Correctly classified chromosome fragments (100% chromosome)

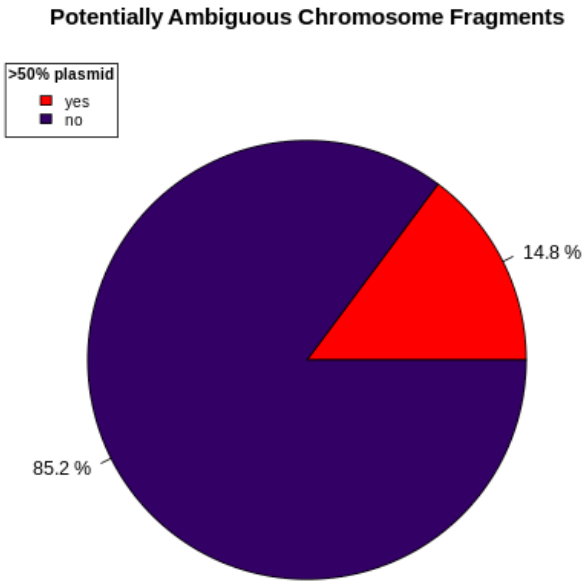

Figure 6.12: Portion of chromosome fragments with >50% plasmid part in potentially ambiguous chromosome fragments (>0% plasmid)

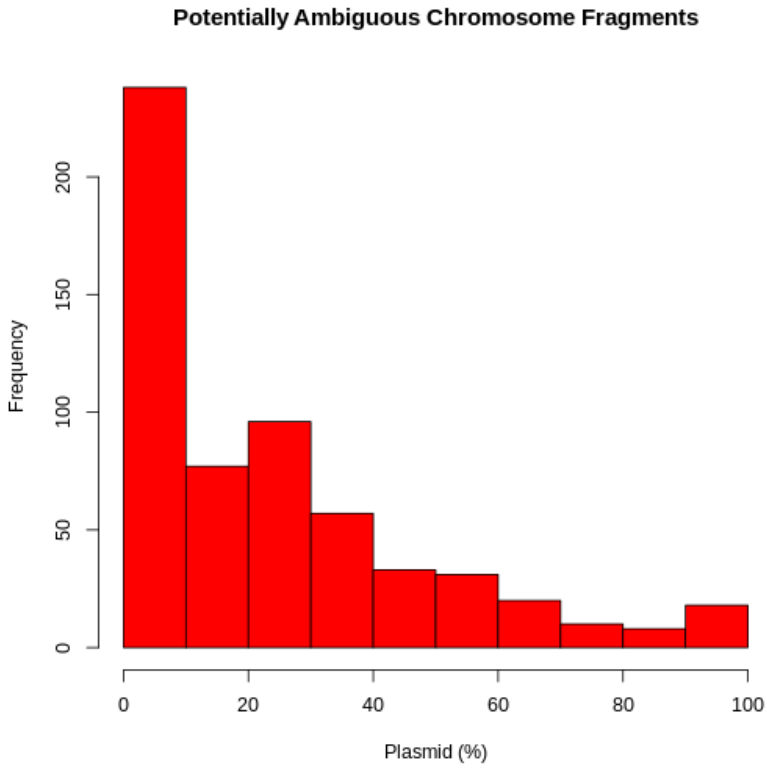

Figure 6.13: Distribution of plasmid portions in potentially ambiguous chromosome fragments

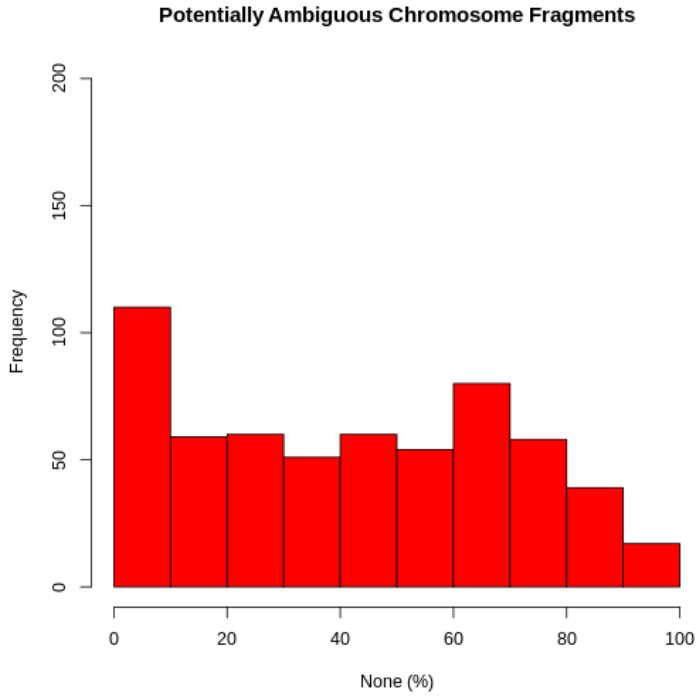

Figure 6.14: Distribution of none portions in potentially ambiguous chromosome fragments

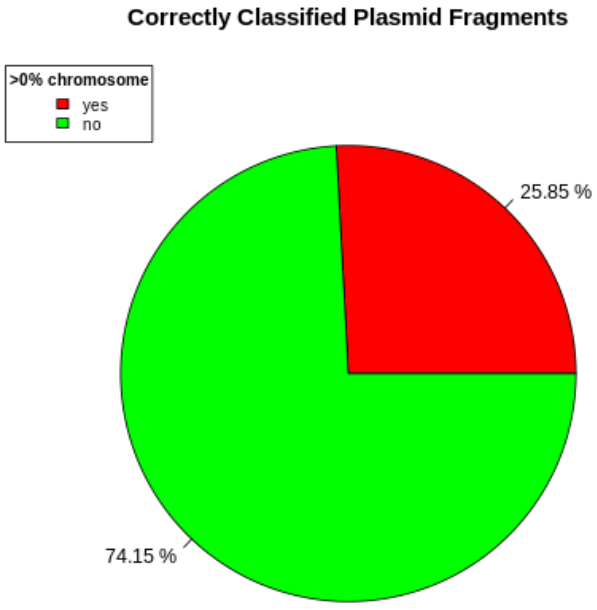

Figure 6.15: Correctly classified plasmid fragments (100% plasmid)

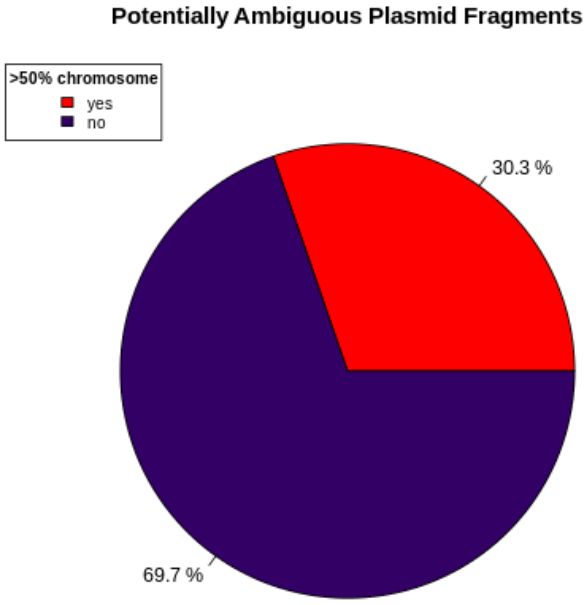

Figure 6.16: Portion of plasmid fragments with >50% chromosome part in potentially ambiguous plasmid fragments (>0% chromosome)

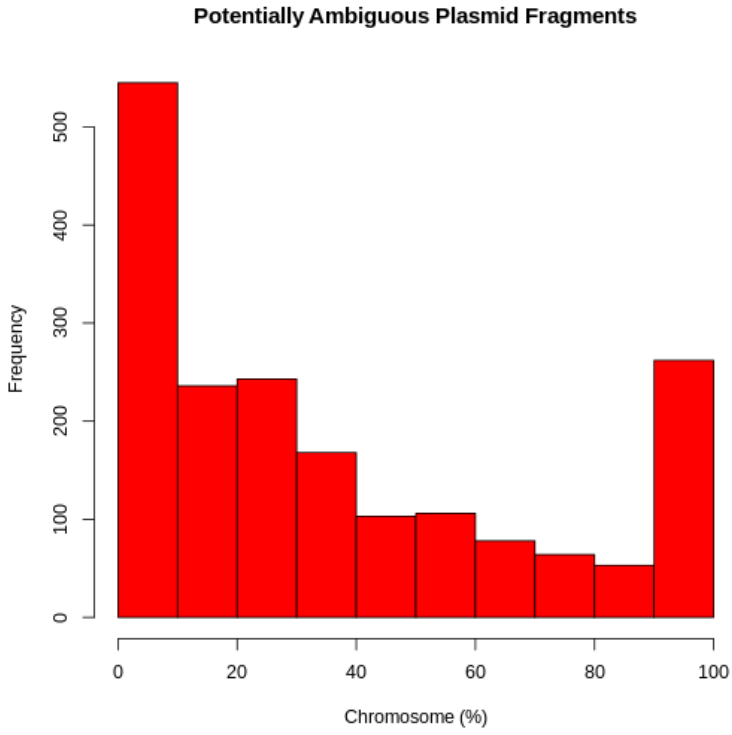

Figure 6.17: Distribution of chromosome portions in potentially ambiguous plasmid fragments

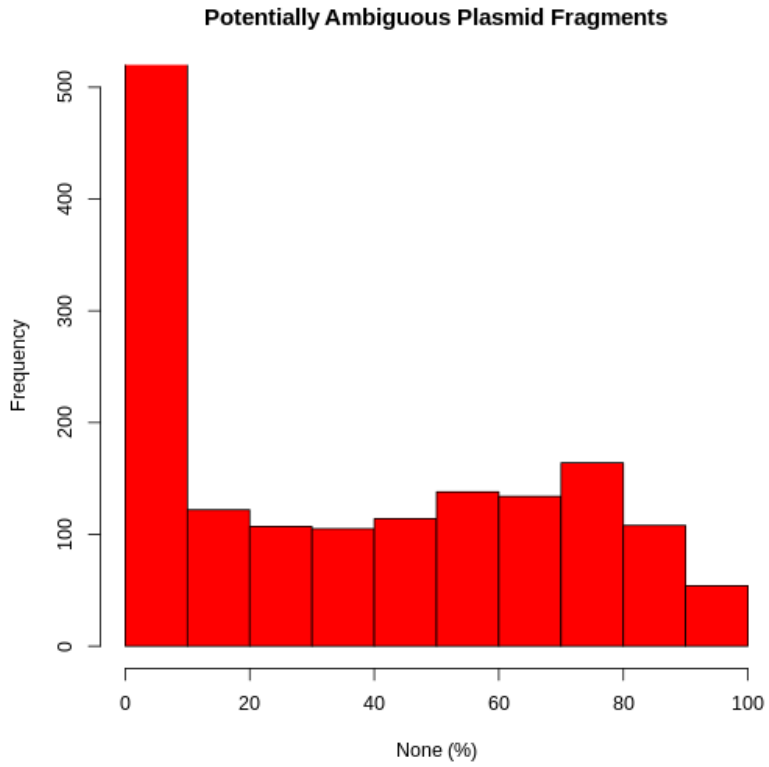

Figure 6.18: Distribution of none portions in potentially ambiguous plasmid fragments

Next, in attempt to plot the ROC curve for plasmid prediction, true positive, false positive, true negative and false negative values are required. On the one hand, every known plasmid fragment P_i prediction consists of chromosome (c_i), plasmid (p_i) and none (n_i) percentage, so that the following equation is valid: $c_i + p_i + n_i = 100\%$. Next step is to define the threshold X_P for plasmid classification and claim that the classification of P_i is correct if p_i is greater or equal to X_P and that P_i is indeed a plasmid, whereas if p_i is lower than X_P we consider P_i to be a misclassified fragment. On the other hand, the same procedure is carried out for each known chromosome fragment C_j, by defining the threshold X_C, used as the cutoff for c_j, in order to provide chromosome classification.

True positive value (TP) is percentage of plasmid fragments (P_i) classified as plasmid, while false positive value (FP) is percentage of chromosome fragments (C_j) also classified as plasmid. In Figure 6.19 true positive (red line) and false positive (blue line) are plotted against threshold X_P. Furthermore, true negative rate (TN) is

percentage of chromosome fragments (C_j) classified as chromosome, whilst false negative rate (FN) is percentage of plasmid fragments (P_i) classified as chromosome as well. In Figure 6.20 true negative rate (blue line) and false negative rate (red line) are plotted against threshold X_C.

In conclusion, lines in Figure 6.19 and Figure 6.20 are monotonically decreasing, i.e. fewer fragments meet the cutoff, while X_P and X_C are increasing. It can again be seen that classification for chromosomes is more accurate, with line in Figure 6.20 staying near 100% with the gradually increasing X_C threshold.

Finally, sensitivity (true positive rate) and specificity (true negative rate) are calculated based on formulae (4.1) and (4.2) and ROC curve plotted in Figure 6.21. ROC curve shows that plasmid prediction for 'unknown' plasmids is quite reasonable, even though there are room for some improvements.

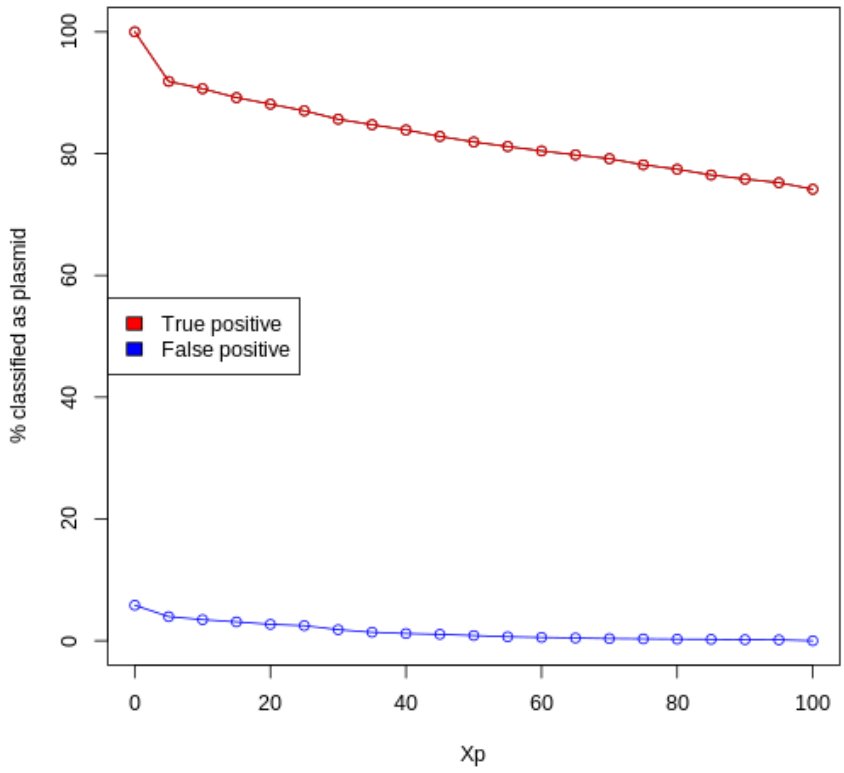

Figure 6.19: True and false positive values for plasmid prediction

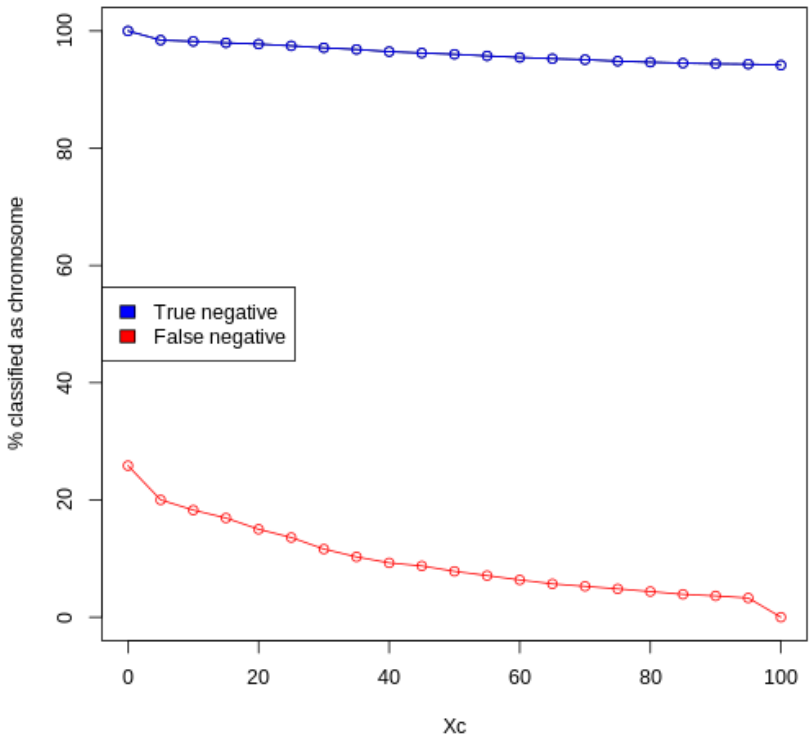

Figure 6.20: True and false negative values for plasmid prediction

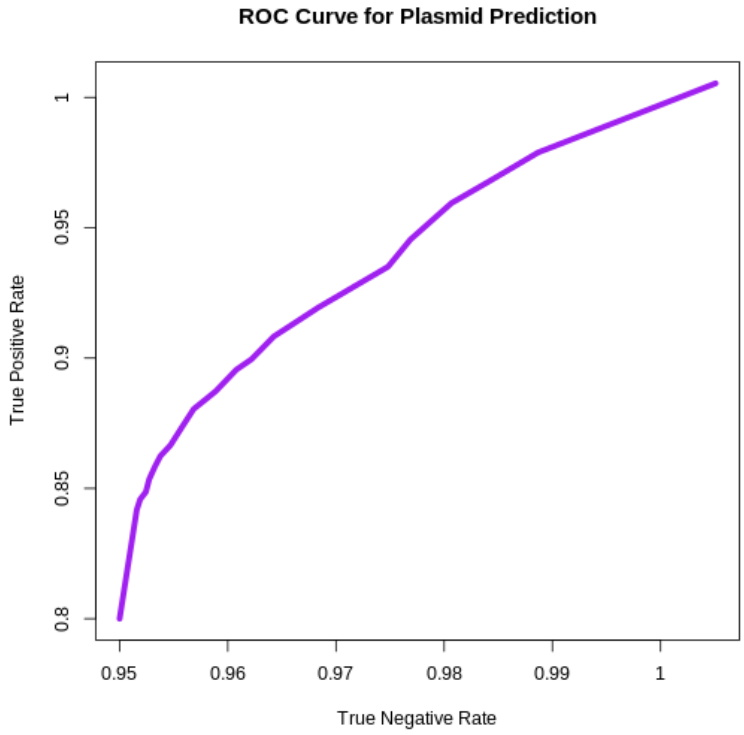

Figure 6.21: ROC curve for *Escherichia coli*

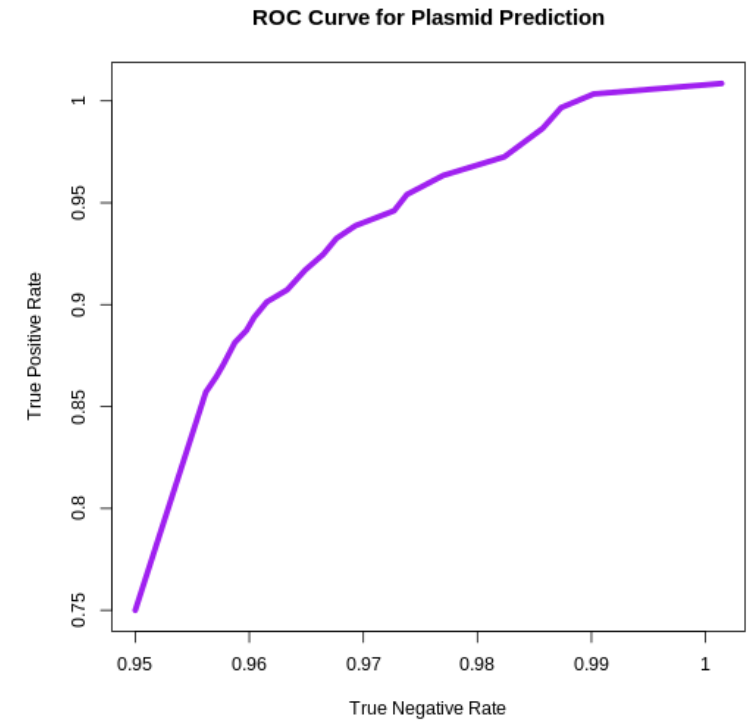

Figure 6.22: ROC curve for *Klebsiella pneumoniae*

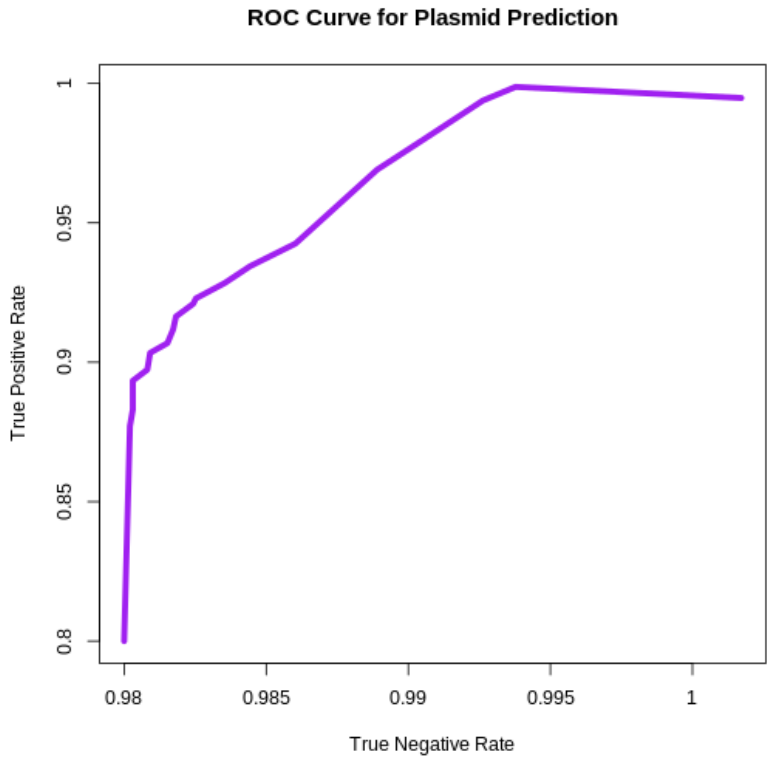

Figure 6.23: ROC curve for *Salmonella enterica*

Through thorough analysis of ROC curves in Figure 6.21, Figure 6.22 and Figure 6.23 and statistics in Table 6.2, it can be determined that classification of *Salmonella enterica* is undoubtedly the best one, particularly for chromosomes. Classification of plasmids for *Escherichia coli* and *Salmonella enterica* is quite similar, whilst the one for *Klebsiella pneumoniae* shows the worst results.

All things considered, classification of unknown chromosomes and plasmids is expectedly worse than the one shown in Table 6.1, but it is still quite successful, especially when taking into consideration the fact that the classification is conducted on queries that have not been seen before, i.e. are not contained in the database.

Table 6.2: Potentially ambiguous chromosome/plasmid fragments for three bacterial species

	> 0 % plasmid/chromo	**> 25 % plasmid/chromo**	**> 50 % plasmid/chromo**	**> 75 % plasmid/chromo**
Chromosome, *Escherichia coli*	5.83 %	2.48 %	0.86 %	0.31 %
Chromosome, *Klebsiella pneumoniae*	5.42 %	2.71 %	1.62 %	0.94 %
Chromosome, *Salmonella enterica*	2.22 %	0.60 %	0.18 %	0.03 %
Plasmid, *Escherichia coli*	25.85 %	13.59 %	7.83 %	4.84 %
Plasmid, *Klebsiella pneumoniae*	34.87 %	22.65 %	16.62 %	11.62 %
Plasmid, *Salmonella enterica*	24.19 %	13.91 %	10.66 %	8.11 %

6.3. Implementation comparison

The above-mentioned results are obtained using the BLAST method. There is no reason to repeat all of the visualisations and graphs for this thesis' method, because they are quite similar, so comparison is going to be reported in percentages.

First of all, execution speed of classification method using algorithm implemented in this thesis was in average 27% greater than whilst using existing BLAST method. For comparison, classification of hundred randomly chosen chromosomes with BLAST method lasted approximately 26 hours, while classification with this thesis' algorithm lasted approximately 18.5 hours. Furthermore, duration of classification for whole dataset of plasmids, using the BLAST method, is roughly 4 whole days, whereas the implementation, using minimap2 algorithm, lasted circa 3 days.

Secondly, alignment scores for thesis' algorithm are worse than BLAST algorithm, especially for far away hits, which consequently leads to inferior classification results. Results of chromosome classification are in average 5.79% worse, i.e. more chromosomes get misclassified or unclassified. It is important to state that tests are conveyed on hundred randomly chosen chromosomes because of their immense size and thus large time consumption of classification process. In average chromosomes have 864 fragments, and with them being the length of 5 kpb, average size of chromosome is 4320 kpb. Furthermore, plasmid classification is 6.3% worse, and they tend to get misclassified as chromosome, rather than to be unclassified.

In conclusion, implemented method, using minimap2, is much more effective considering time consumption, but gives poorer results while conducting the classification. Future work should include improvements of the alignments, i.e. sequence similarity comparisons, particularly for far away hits, whilst still not sacrificing computing efficiency if feasible.

7. Conclusion

The main concern of this thesis is analysis of a reference based plasmid prediction algorithm for short read assemblies based on BLAST, and attempt at developing novel plasmid prediction algorithm which is based on minimap2. Existing BLAST algorithm demonstrates better classification results, whilst the thesis' algorithm based on minimap2 proved to have shorter execution time.

Classification is conducted on both known and unknown plasmids and chromosomes, which are divided in k-length fragments in order to get the illusion of incomplete assembly. Furthermore, cross-validation is carried out for the three most frequent and clinically important bacterial species in database: *Escherichia Coli, Klebsiella Pneumoniae* and *Salmonella enterica.* The results are expectedly worse than the ones for known species, but are still proven to be quite reasonable.

To conclude, classification results are mainly accurate, but there is still room for improvement. Described algorithms and automatization of plasmid prediction in general have a great potential, particularly with the increase of known plasmids in databases.

References

[1] S. Arredondo-Alonso, R. J. Willems, W. van Schaik, A. C. Schürch. *On the (im)possibility of reconstructing plasmids from whole genome short-read sequencing data*. Microbial Genomics. 2017.

[2] J. Brennan. (2018). *The Difference Between Genomic DNA & Plasmid DNA.* Source: https://sciencing.com/difference-between-genomic-dna-plasmid-dna-2314.html.

[3] Science Learning Hub – Pokapū Akoranga Pūtaiao. (2014). *Bacterial DNA – the role of plasmids*. Source: https://www.sciencelearn.org.nz/resources/1900-bacterial-dna-the-role-of-plasmids.

[4] E. Kudirkiene, L. A. Andoh, S. Ahmed, A. Herrero-Fresno, A. Dalsgaard, K. Obiri-Danso, J. E. Olsen. *The Use of a Combined Bioinformatics Approach to Locate Antibiotic Resistance Genes on Plasmids From Whole Genome Sequences of Salmonella enterica Serovars From Humans in Ghana.* Frontiers in Microbiology. 2018.

[5] L. Brooks, M. Kaze, M. Sistrom, *J. C. Dunning Hotopp. A Curated, Comprehensive Database of Plasmid Sequences.* Microbiology Resource Announcements. 2019.

[6] S. Delaney, R. Murphy, F. Walsh. *A Comparison of Methods for the Extraction of Plasmids Capable of Conferring Antibiotic Resistance in a Human Pathogen From Complex Broiler Cecal Samples.* Frontiers in Microbiology. 2018.

[7] S. Arredondo-Alonso, M. R. C. Rogers, J. C. Braat, T. D. Verschuuren, J. Top, J. Corander, R. J. L. Willems, A. C. Schürch. *mlplasmids: a user-friendly tool to predict plasmid- and chromosome-derived sequences for single species.* Microbial Genomics. 2018.

[8] Wikipedia. (2019). *Bacteria.* Source: https://en.wikipedia.org/wiki/Bacteria.

[9] T. F. Jesus, B. Ribeiro-Gonçalves, D. N. Silva, V. Bortolaia, M. Ramirez, J. A. Carriço. *Plasmid ATLAS: plasmid visual analytics and identification in high-throughput sequencing data*. Nucleic Acids Research. Volume 47, 2019. [5]

[10] Wikipedia. (2019). *Plasmid.* Source: https://en.wikipedia.org/wiki/Plasmid.

[11] Online Textbook: Chapter 13. *Frontiers of Genetics: Biologists can engineer bacteria to make useful products*. Figure 13-3. Source: http://bodell.mtchs.org/OnlineBio/BIOCD/text/chapter13/concept13.2.html

[12] Houghton Mifflin Harcourt. (2016). *The Bacterial Chromosome and Plasmid*. Source: https://www.cliffsnotes.com/study-guides/biology/microbiology/microbial-genetics/the-bacterial-chromosome-and-plasmid

[13] P. Bradley, H. C. den Bakker, E. P. C. Rocha, G. McVean, Z. Iqbal. *Ultrafast search of all deposited bacterial and viral genomic data.* Nature Biotechnology. 2019.

[14] G. Royer, J. W. Decousser, C. Branger, M. Dubois, C. Médigue, E. Denamur, D. Vallenet. *PlaScope: a targeted approach to assess the plasmidome from genome assemblies at the species level.* Microbial Genomics. 2018.

[15] P. S. Krawczyk, L. Lipinski, A. Dziembowski. *PlasFlow: predicting plasmid sequences in metagenomic data using genome signatures*. Nucleic Acids Research. Vol. 46, No. 6, 2018.

[16] A. J. Page, A. Wailan, Y. Shao, K. Judge, G. Dougan, E. J. Klemm, N. R. Thomson, J. A. Keane. *PlasmidTron: assembling the cause of phenotypes and genotypes from NGS data.* Microbial Genomics. 2018.

[17] J. Robertson, J. H. E. Nash. *MOB-suite: software tools for clustering, reconstruction and typing of plasmids from draft assemblies*. Microbial Genomics. 2018.

[18] G. Drakos. (2018). *Cross-Validation.* Source: https://towardsdatascience.com/cross-validation-70289113a072.

[19] R. Khandelwal. (2018). *K fold and other cross-validation techniques.* Source: https://medium.com/datadriveninvestor/k-fold-and-other-cross-validation-techniques-6c03a2563f1e.

[20] S. Narkhede. (2018). *Understanding AUC - ROC Curve.* Source: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

[21] U.S. National Library of Medicine. *Basic Local Alignment Search Tool.* Source: https://blast.ncbi.nlm.nih.gov/Blast.cgi.

[22] Wikipedia. (2019). *BLAST.* Source: https://en.wikipedia.org/wiki/BLAST.

[23] H. Li. *Minimap2: pairwise alignment for nucleotide sequences*. Bioinformatics. Volume 34, Issue 18, 2018.

Using Reference Database for Plasmid Prediction

Abstract

The main concern of this thesis is analysis of a reference based plasmid prediction algorithm for short read assemblies based on BLAST, and development of novel algorithm based on minimap2. Even though it has shorter execution time, thesis' algorithm demonstrates poorer classification results than existing BLAST algorithm, so future improvements are needed. Classification is conducted on both known and unknown plasmids and chromosomes, which are parted in k-length subsequences in order to mimic incomplete assemblies. Furthermore, cross-validation is carried out for the three most frequent and clinically important bacterial species in database: *Escherichia Coli, Klebsiella Pneumoniae* and *Salmonella enterica*. Source code is available at: https://github.com/lbcb-edu/BSc-thesis-18-19/tree/sdeur.

Keywords: bacterium, plasmid, chromosome, classification, cross-validation

Korištenje referentne baze za predviđanje plazmida

Sažetak

Glavna zadaća ovoga rada jest analiza algoritma zaduženog za predviđanje plazmida na temelju referentne baze za kratka očitanja, temeljenog na BLAST-u, te razvoj novog algoritma temeljenog na algoritmu minimap2. Unatoč tome što ima kraće vrijeme izvođenja, algoritam razvijen u ovom radu pokazuje slabije rezultate klasifikacije, nego postojeći BLAST algoritam, stoga su potrebna buduća poboljšanja. Klasifikacija se provodi i na poznatim i na nepoznatim plazmidima i kromosomima, koji su podijeljeni na podnizove duljine k. Nadalje, unakrsna validacija provodi se za tri najučestalije i medicinski najbitnije vrste bakterija u bazi, sljedećih naziva: *Escherichia Coli, Klebsiella Pneumoniae* i *Salmonella enterica*. Izvorni kod dostupan je na: https://github.com/lbcb-edu/BSc-thesis-18-19/tree/sdeur.

Ključne riječi: bakterija, plazmid, kromosom, klasifikacija, unakrsna validacija