

# **A COMPUTER SIMULATION OF PLASMID EVOLUTION IN BACTERIA**

A Thesis

Presented in Partial Fulfillment of the Requirements for the

Degree of Master of Science

with a

Major in Bioinformatics and Computational Biology

in the

Department of Mathematics

College of Graduate Studies

University of Idaho

by

**Allison M. Tucker**

August 2014

Major Professor: Eva M. Top, Ph.D.

## Authorization to Submit Thesis

This thesis of Allison Tucker, submitted for the degree of Master of Science with a Major in Bioinformatics and Computational Biology and titled "A Computer Simulation of Plasmid Evolution in Bacteria," has been reviewed in final form. Permission, as indicated by the signatures and dates below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor: \_\_\_\_\_ Date: \_\_\_\_\_  
Eva Top, Ph.D.

Committee  
Members: \_\_\_\_\_ Date: \_\_\_\_\_  
Stephen Krone, Ph.D.

\_\_\_\_\_ Date: \_\_\_\_\_  
José Miguel Ponciano Castellanos, Ph.D.

\_\_\_\_\_ Date: \_\_\_\_\_  
Terence Soule, Ph.D.

Department  
Administrator: \_\_\_\_\_ Date: \_\_\_\_\_  
Eva Top, Ph.D.

College Dean: \_\_\_\_\_ Date: \_\_\_\_\_  
Paul Joyce, Ph.D.

### Final Approval and Acceptance

Dean of the College  
Of Graduate Studies: \_\_\_\_\_ Date: \_\_\_\_\_  
Jie Chen, Ph.D.

## **Abstract**

Bacterial plasmids are extra-chromosomal, typically circular DNA elements that require a bacterial host to survive, but replicate separately from the host chromosome. Plasmids may confer host-beneficial traits to and co-evolve with their hosts, yet many of the underlying genetic mechanisms of plasmid-host adaptation are still unknown. The work highlighted here revolved around the creation of a computer model to simulate plasmid evolution in bacterial hosts and observe the underlying mechanisms of adaptation and the impact of different plasmid characteristics on population diversity. This was done by changing features of the ancestral plasmid (i.e. cost, stability, and transferability and mutations rates) and monitoring the effect on the population over time. The output of the simulation was comparable to results of published experimental evolution studies. This model can now be used to help explain experimentally observed plasmid evolution dynamics and generate hypotheses about plasmid evolution that may later be tested empirically.

## Acknowledgements

I would like to thank my advisor, Eva Top, for her direction, support, and inspiration during my studies. I would also like to thank my previous advisor, Zaid Abdo, for his initiation of this project. Additionally, I'd like to show my gratitude towards my committee members, Steve Krone, José Ponciano, and Terry Soule, for their help and advice. Furthermore, Wesley Loftie-Eaton provided much help in every aspect of this thesis and for that I am extremely grateful. I'd also like to show my appreciation to members of the Top lab – including Diya Sen, Hirokazu Yano, Sylvia Smith, and Linda Rogers – for their encouragement throughout this project. Further gratitude is given to the rest of the Bioinformatics and Computational Biology faculty, staff, and students, for their friendship and backing for the course of my studies. Finally, I'd like to give my sincerest thanks to my family and friends, especially my husband, for their unending support, encouragement, love, and understanding. Thank you to everyone who helped in any way during this time.

## **Dedication**

I would like to dedicate this thesis to my husband, Seth Thompson, for the love, support, and patience he provided me with during my pursuit of this degree.

## Table of Contents

<b>Authorization to Submit Thesis.....</b>	<b>ii</b>
<b>Abstract.....</b>	<b>iii</b>
<b>Acknowledgements .....</b>	<b>iv</b>
<b>Dedication .....</b>	<b>v</b>
<b>Table of Contents .....</b>	<b>vi</b>
<b>List of Figures .....</b>	<b>viii</b>
<b>List of Tables.....</b>	<b>ix</b>
<b>Chapter I: Introduction .....</b>	<b>1</b>
<b>1.1 Background .....</b>	<b>1</b>
<b>1.2 History.....</b>	<b>2</b>
<b>1.3 Plasmid Structure .....</b>	<b>2</b>
<b>1.4 Replication .....</b>	<b>3</b>
<b>1.5 Conjugation .....</b>	<b>4</b>
<b>1.6 Maintenance Mechanisms.....</b>	<b>5</b>
Copy Number.....	5
Multimer Resolution.....	6
Active Partitioning .....	7
Toxin/Antitoxin .....	7
Plasmid stability.....	8
<b>1.7 Accessory Region.....</b>	<b>8</b>
<b>1.8 Antibiotic Resistance .....</b>	<b>9</b>
<b>1.9 Plasmid Evolution and Coevolution .....</b>	<b>10</b>
<b>1.10 Computer Simulations .....</b>	<b>11</b>
<b>1.11 Goals .....</b>	<b>12</b>

<b>Chapter 2: Simulation</b> .....	<b>14</b>
<b>2.1 Input Parameters</b> .....	<b>14</b>
<b>2.2 Assumptions</b> .....	<b>19</b>
<b>2.3 Simulation</b> .....	<b>20</b>
<b>2.4 Output</b> .....	<b>24</b>
<b>2.5 Running the Simulation</b> .....	<b>25</b>
<b>Chapter 3: Results</b> .....	<b>26</b>
<b>3.1 Simulation Parameters and Values</b> .....	<b>26</b>
<b>3.2 Running the Simulation</b> .....	<b>29</b>
<b>3.3 Mutation Rate</b> .....	<b>31</b>
<b>3.4 Conjugation Rate</b> .....	<b>33</b>
<b>3.5 Loss Rate</b> .....	<b>35</b>
<b>3.6 Cost</b> .....	<b>37</b>
<b>3.7 Principal Component Analysis</b> .....	<b>39</b>
<b>3.8 Validation</b> .....	<b>46</b>
<b>Chapter 4: Discussion, Applications, and Future Work</b> .....	<b>51</b>
<b>4.1 Discussion</b> .....	<b>51</b>
<b>4.2 Applications</b> .....	<b>52</b>
<b>4.3 Future Work</b> .....	<b>53</b>
<b>References</b> .....	<b>55</b>
<b>Appendix</b> .....	<b>61</b>
<b>Code</b> .....	<b>61</b>

## List of Figures

Figure 1: Schematic depiction of plasmid structure .....	3
Figure 2: Model for theta replication of a plasmid .....	4
Figure 3: Plasmid multimer resolution system .....	7
Figure 4: Steps of the simulation .....	21
Figure 5 Example of the information kept for each host .....	25
Figure 6 Differences in evolutionary population dynamics due to parameters.....	31
Figure 7: Mutation rate comparison .....	33
Figure 8: Conjugation rate comparison.....	35
Figure 9: Loss rate comparison .....	37
Figure 10: Cost comparison .....	39
Figure 11: Scatterplot of correlation of statistics.....	43
Figure 12: Plot of first two principal components against each other .....	45
Figure 13: Percent of total mutations stemming from each mutation region.....	46
Figure 14: Simulation emulating De Gelder <i>et al.</i> (2007).....	48
Figure 15: Simulation emulating Hughes <i>et al.</i> (2012).....	50



## List of Tables

Table 1 Simulation parameters and values .....	27
Table 2 Variable parameters and their values .....	28
Table 3 Proportion of variance for all principal components .....	42

## Chapter I: Introduction

### 1.1 Background

Plasmids are extra-chromosomal, typically circular DNA molecules (Roth and Helinski, 1967). Though they are physically separate from, and can replicate independently of, chromosomal DNA within cells, plasmids do require a host in order to replicate and often provide a selective advantage to that host. Many plasmids are able to transfer horizontally, occasionally to a wide range of genetically distinct bacteria (del Solar *et al.*, 1996; De Gelder *et al.*, 2008). These so-called broad-host-range (BHR) plasmids are capable of acting as mobile genetic elements, spreading different plasmid-encoded traits to numerous phylogenetic groups of bacteria (Sen *et al.*, 2011).

Plasmids are able to confer traits to their hosts to allow for host (and plasmid) survival in hostile environments (Kado, 1998). Such traits can include, but are not limited to: resistance to antibiotics (Foster, 1983; Thavasi *et al.*, 2004; Canton, 2009; Lawson, 2008; Carattoli, 2009), fosfomycin (Suarez and Mendoza, 1991), quinolone (Strahilevitz *et al.*, 2009), trimethoprim (Huovinen, 1987), and heavy metals (Foster, 1983); colicin production and immunity (Dougan *et al.*, 1978; Spangler *et al.*, 1985); EcoRI restriction modification systems (Betlach *et al.*, 1976; Rimšlienė *et al.*, 1995); a plasmid-borne rRNA operon (Kunnimalaiyaan *et al.*, 2001); catabolism of herbicides and aromatic pollutants such as atrazine (Martinez *et al.*, 2001), toluene (Martínez-Lavanchy *et al.*, 2010), and methyl parathion and *p*-nitrophenol (Liu *et al.*, 2005); and the synthesis of exotoxins (Lowe and Glomski, 2012; Hernández *et al.*, 2003). These traits may provide a plasmid's host with a selective advantage under certain environmental conditions. Plasmids can also have a negative effect on their hosts, typically in the form of imposing a cost to their hosts and thereby lowering the hosts' overall fitness and competitiveness (Slater *et al.*, 2007). If a plasmid confers a great cost to its host, the host will not be able to successfully compete in the population or community. This can lead to a removal of the host's genes, and subsequently the plasmid, from the population.

## 1.2 History

The term 'plasmid' was introduced in 1952 by Joshua Lederberg as a "generic term for any extra-chromosomal hereditary determinant" (Lederberg, 1952); however, for over a decade, the term was seldom used in scientific papers and rarely appeared in the titles of articles. It wasn't until 1961 (Coe, 1961) that this term began to appear more often and started to specifically refer to a category of circular DNA found in bacteria (Lederberg, 1998). Plasmid research surged so much in the following decade that a journal dedicated to plasmid research, entitled *Plasmid*, was created in 1977 (Plasmid, 1977). The field has continued to grow and in 1998, Lederberg noted, "about 3000 articles a year are published with 'plasmid' in the title or abstract". Some of the first discoveries included a plasmid's ability to synthesize penicillinase (Novick and Richmond, 1965), formation of plasmid heterodiploids (Richmond, 1965), the bacterial plasmid colicinogenic factor E<sub>1</sub> (Roth and Helinski, 1967), and plasmid-encoded resistance to multiple antibiotics (Sompolinsky *et al.*, 1967). Since that time, volumes of data have been compiled on plasmid behavior, manipulation, as well as medical and environmental effects.

## 1.3 Plasmid Structure

Plasmids have two main components to their structure – the backbone and accessory regions (Norman *et al.*, 2009). The backbone region (seen in blue, red, and green in Figure 1) consists of everything the plasmid needs in order to survive; it encodes the replication mechanism, maintenance (stability), and, if self-transmissible, conjugative transfer of the plasmid. Further explained below, maintenance mechanisms can include toxin-antitoxin, copy number, multimer resolution, and active partitioning systems (see Section 1.6). The accessory region (seen in orange in Figure 1) of the plasmid is "extra", meaning that these genes are not strictly necessary for the plasmid's survival, but can be beneficial or even essential to their host in certain environments (Norman *et al.*, 2009). Specifically, genes related to resistance or other host-beneficial traits listed above are found in the accessory region.

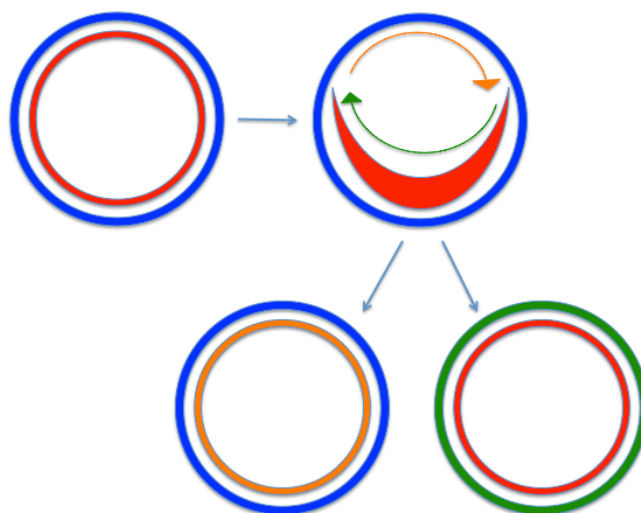


**Figure 1:** Schematic depiction of plasmid structure. The order of the blocks can differ and accessory genes may be interspersed between blocks of backbone genes.

## 1.4 Replication

Plasmids can be limited in the range of bacteria they are capable of replicating in. Plasmids, like viruses, replicate independently of their host, but require the use of their hosts' replication mechanisms (Novick, 1987). There are three general mechanisms for plasmid replication – theta type, strand displacement, and rolling circle (del Solar *et al.*, 1998). Theta-type replication (an example of which can be seen in Figure 2) is widespread amongst plasmids in Gram-negative bacteria. DNA replication through this mechanism can start from one or more origins and can proceed either uni- or bidirectionally. Strand displacement replication requires three plasmid-encoded proteins for initiation of DNA replication and may proceed in either direction. Rolling circle replication is generally found in plasmids from Gram-positive bacteria. DNA replication under this mechanism is unidirectional and the leading strand and synthesis of the lagging strand are uncoupled. Plasmid replication heavily depends on the successful interaction of plasmid DNA and proteins with host factors. Once properly interacting, the double-stranded plasmid DNA will be separated, allowing access to host enzymes – including an initiator protein (DnaA),

helicase (DnaB), helicase accessory protein (DnaC), primase (DnaG), gyrase, and multiple DNA polymerases (Khan, 1997). Following the correct positioning of these enzymes, DNA is further unwound by helicase, exposing single stranded DNA for replication to proceed. Termination of the process results in two plasmids, each containing one parental and one daughter strand of DNA (Scott, 1984). Inheritance of genetic material from a parent host is known as vertical transmission.



**Figure 2: Model for theta replication of a plasmid. The initial two (blue and red) strands of the plasmid are separated and replication proceeds, creating new daughter strands (orange and green) to complement the original parental strands.**

Plasmids are classified by which replication mechanisms (and, therefore, host factors) they utilize. Two plasmids using the same replication mechanisms cannot both survive in a single cell (Novick, 1987). Eventually only one plasmid will successfully be passed on to the host's daughter cells. This is termed incompatibility. Plasmids of the same incompatibility group will compete for the host's factors and resources. This competition can result in the loss of a plasmid. Loss occurs either through competition (with one plasmid dominating over the other) or by simple random chance.

## 1.5 Conjugation

Several different types of plasmids are able to transfer to new host cells through conjugation. When a plasmid-bearing cell (the donor) comes into direct contact with

another (recipient) cell, DNA transfer is enabled between the two. Conjugative plasmids of Gram-negative bacteria encode four major components – an origin of transfer, a relaxase, a type IV coupling protein, and a type IV secretion system (T4SS) (Smillie *et al.*, 2010). Encoded enzymes facilitate processing the DNA transfer while mating pair formation genes code for the pilus and conjugative pore formation, allowing transfer and maintaining contact between cells. Conjugation then proceeds through synthesis of a T4SS (Norman *et al.*, 2009). After the pilus is connected, the relaxase mediates nicking of the plasmid at the origin of transfer, allowing a single strand of plasmid DNA to transfer from donor to recipient. A complimentary strand is then synthesized in both cells. Inheritance of genetic material through conjugation is known as horizontal transmission.

## **1.6 Maintenance Mechanisms**

Even if a plasmid is successfully able to replicate in a bacterial host, it still may fail to be inherited over time. Many plasmids do not always replicate and/or segregate perfectly and, as a result, daughter cells are often left without plasmids. Thus plasmids utilize a number of mechanisms to improve the stable inheritance of plasmids through vertical transmission (Fernández-López *et al.*, 2006). These methods include: a) copy number control, b) multimer resolution, c) active partitioning, and d) toxin/antitoxin systems. The following paragraphs highlight each of these mechanisms.

### **Copy Number**

Increasing the number of plasmid copies per cell is one method used by plasmids to decrease segregational loss. As the number of plasmid copies in a cell increases, so do the chances that each daughter cell will retain at least one copy (Schmidt *et al.*, 1996). This increase in number can be attained by relaxation of the copy number control system or an increase in plasmid replication efficiency – a plasmid that replicates faster than its host will result in multiple copies of the plasmid residing in the cell. A larger number of plasmid copies, however, also results in an increased cost to the host, which can result in the host being less fit in the population. A host that is unable to replicate as quickly, due to a costly

plasmid, will likely be out-competed by other hosts and will fail to maintain a presence in the population. Therefore it is vital that an optimal number of plasmid copies is maintained in the host; too few and the plasmid may be lost, too many and the host's replication rate will be adversely affected.

There are three general classes of copy number control mechanisms – those with directly repeated sequences and initiator, those using only antisense RNAs, and those involving an antisense RNA in combination with a protein (del Solar and Espinosa, 2000). The copy number of a given plasmid is usually a fixed range (with an upper bound where replication is unfavored and a lower bound where it is favored) in a certain host and under certain growth conditions, but can also be highly dynamic. Mutations in this region can have a myriad of consequences for both the host and plasmid.

### **Multimer Resolution**

Another event of plasmid loss is through dimers. A dimer is essentially the “intertwining” of two plasmids; two plasmids joined together in such a way that reduces the number of plasmids available during segregation; see Figure 3. Dimers can form through recombination or replication and result in a decrease in the effective number of plasmids to be distributed between daughter cells (Sengupta and Austin, 2011). Plasmid multimer resolution systems are used to counteract plasmid dimers. This involves different types of genes depending on the type of plasmid, which convert dimerized plasmids to monomers (Summers, 1998).

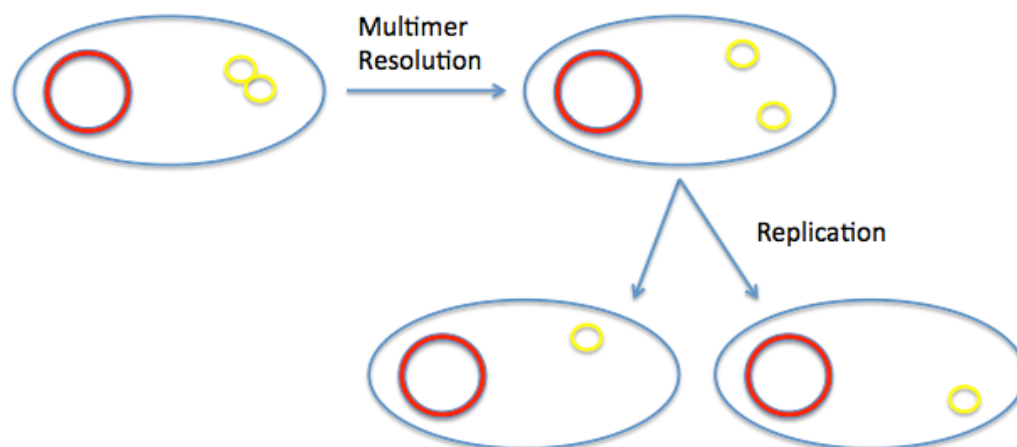


Figure 3: Plasmid multimer resolution system

### Active Partitioning

Active partitioning is another mechanism for promoting stable plasmid inheritance. This places plasmid pairs, bound at centromere-like sequences after replication, near the center of the bacterial cell (Zielenkiewicz and Ceglowski, 2001). After replication, the twin daughter molecules are moved and tethered to cell poles (Jensen and Gerdes, 1999). When the cell subsequently splits at its center, the plasmids are distributed to the two daughter cells.

### Toxin/Antitoxin

Finally, toxin/antitoxin systems ensure a plasmid's survival by postsegregational killing or inhibition of plasmid-free cells. There are two different types of toxin-antitoxin systems, one where the antitoxin is an RNA and one where it's a protein (the toxin is always a protein) (Van Melderen and De Blast, 2009). A protein or RNA toxin-antitoxin system encodes two intracellular components, one for the toxin and one for the antitoxin. Toxins thus far characterized include "gyrase inhibitors, phosphotransferases, site-specific ribonucleases, ribosome-dependent ribonucleases, and a possible riboexonuclease" (Magnuson, 2007). The toxin is more stable than the antitoxin, persisting in the cytoplasm



longer and resulting in an inhibition of cell division for any segregant that loses a plasmid. Eliminating plasmid-free cells can be beneficial by reducing the competition for resources for those hosts with plasmids.

### **Plasmid stability**

Plasmid stability is a measure of the likelihood with which a plasmid is retained in a population over time in the absence of positive selection for the plasmid (De Gelder *et al.*, 2007) and is influenced by the maintenance mechanisms described above, the cost of the plasmid, and horizontal transfer. The stability of a plasmid can be used as a phenotype to evaluate the adaptation of plasmids to hosts. Mutations may make a plasmid more or less stable in a host by affecting its maintenance mechanisms, i.e. improving the efficiency of these mechanisms and/or employing more mechanisms to ensure the inheritance of the plasmid, or by lowering the cost of the plasmid, or increasing the transfer efficiency of the plasmid, or a combination of these. A higher transfer efficiency will lead to greater reinfection of hosts that have lost their plasmid. A plasmid's accessory genes can be beneficial to a host in the presence of selection for the beneficial genes and a plasmid can encode for mechanisms to prevent its loss. However, imperfect allocations of plasmids between daughter cells during host replication and the cost of the plasmid imposing a growth disadvantage to its host can lead to a reduction of plasmids in a population. In the absence of selection, the cost of a plasmid to its host will lead to plasmid-free variants out-competing plasmid-bearing cells.

### **1.7 Accessory Region**

The accessory region in a plasmid is a variable region that can encode for numerous host-beneficial traits. This can include resistance to antibiotics, heavy metal resistance, colicin production and immunity, and catabolism of numerous substances, among others, as described above (see Section 1.1).

It is also worth noting that not all plasmids have accessory genes. These are termed "cryptic" plasmids (Hawkey *et al.*, 1984). It is unclear what mechanisms they employ to

ensure their survival in a population or how prevalent these plasmids are in nature, though it is likely that they are more common and diverse than any of the resistance plasmids isolated thus far. One hypothesis is that these plasmids resemble a “parasitic existence through efficient horizontal transfer” (Brown *et al.*, 2013), though more research on these plasmids is needed.

## **1.8 Antibiotic Resistance**

One type of accessory genes that plasmids may carry are resistance genes. These genes greatly assist the host and ensure the plasmid’s survival within a population if selective pressure (i.e. the antibiotic for which resistance is encoded) is present. In the absence of selective pressure, however, these additional (and unnecessary) genes may result in the plasmid being lost from a population, due to the cost of the plasmid to the host and the resultant decline in host fitness.

Antibiotic resistance is becoming a worldwide problem, borne from the adaptive response of bacteria to the widespread use of antibiotics (Levy, 2001). Research complacency has allowed this threat to evolve into an emerging crisis (Spellberg *et al.*, 2008) and more and more resistance to antibiotics has been observed (Palumbi, 2001; World Health Organization, 2014). Bacterial infections resistant to many antibiotics are becoming increasingly common and present a worldwide threat. A post-antibiotic era “in which common infections and minor injuries can kill” (World Health Organization, 2014) is becoming a serious possibility.

Bacteria can become antibiotic resistant as a result of a chromosomal mutation or horizontal gene transfer (Neu, 1992). Horizontal gene transfer (HGT) is the transmission of genetic material between organisms. Unlike vertical gene transfer, which occurs when a cell passes genetic material to its offspring, horizontal gene transfer can occur between two non-related organisms. There are three mechanisms by which HGT can occur - transduction, transformation, and conjugation (Frost *et al.*, 2005). Transduction is the transfer of genetic information from one organism to another through the use of a vector,

such as a virus. Transformation is the uptake and incorporation of genetic material from a cell's surroundings. Conjugation is the transfer of genetic material through direct contact between two cells, typically mediated by conjugative plasmids.

Acquisition of new genes through HGT can occur through mobile genetic elements (MGEs) - pieces of DNA that facilitate the movement of DNA between bacterial cells. Plasmids, viruses, bacteriophages, and transposons are all types of MGEs. Elements like these help to enhance the dynamic nature of the supergenome (Norman *et al.*, 2009).

Transposons are able to insert themselves into different DNA sequences. As a result, one method of transportation for them is to jump onto plasmids or bacteriophages and transfer to a new bacterial cell. Additionally, some transposons, called conjugative transposons, are able to move between cells on their own. Bacteriophages are able to transduce host DNA. Like viruses, some bacteriophages can become virulent, replicating vigorously and causing the host cell to lyse (Frost *et al.*, 2005).

Plasmids, in contrast to bacteriophages, do not kill the host cell after replication. Instead, they replicate with the aim of being passed on to the cell's offspring. Through this vertical and horizontal gene transfer, plasmids can readily become fixed pieces in a population. It is therefore important to study plasmids and their role in bacterial resistance.

## **1.9 Plasmid Evolution and Coevolution**

Plasmids have been shown to evolve to be better adapted to their hosts (Modi and Adams, 1991) and some of the genetic bases for improved stability by plasmid-encoded mutations have been determined. There are a number of different mutations that may occur in a plasmid and encode for improved stability. It can be achieved by decreasing the cost of the plasmid to the host, improved partitioning of a plasmid to daughter cells, increased conjugation and re-infection rates, post-segregational killing mechanisms, and copy number regulation (Sota *et al.*, 2010).

Additionally, it has been shown that both bacteria and plasmids may mutate to increase the stability of the plasmid (Hughes, 2011). Ancestral (wild-type) plasmids, initially unstable

in a bacterial host in the absence of selection, have been evolved under heavy selective pressure. The evolved plasmid was more stable in the evolved host. Experiments were performed to determine if this evolution occurred in either the plasmid or host by combining ancestral hosts with evolved plasmids and evolved hosts with ancestral plasmids. These combinations, while showing slight improvement over the original ancestral host and ancestral plasmid combination, still were significantly less stable than the evolved host and evolved plasmid combination (Sota *et al.*, 2010; Hughes, 2011). This proves that bacteria-plasmid coevolution does occur, allowing for a faster adaptation period to new environments.

Lenski *et al.* (1994) also showed that coevolution occurred – the plasmid pACYC184 reduced the fitness of naïve host genotype but significantly increased the fitness of the coevolved host's genotype. Similarly, Dahlberg and Chao (2003) showed evidence of coevolution in their study. After 1100 generations, the plasmid was able to evolve to no longer impose a cost on the host and the host was able to reduce the impact of the ancestral plasmid's cost on the host's fitness. Additionally, De Gelder *et al.* (2008) showed that a plasmid could evolve to adapt to an unfavorable host. This experimental evolution study also showed that regular switching between distinct hosts hampered plasmid evolution. Dionisio *et al.* (2005) provided evidence that mutations in plasmids have pleiotropic effects on chromosomal changes. Furthermore, their study found a fitness-increasing effect of the evolved plasmid in another species, which has important implications in the maintenance and spread of plasmids. Sota *et al.* (2010) also showed evidence of coevolution in their experiments. Additionally, they also found evolution that lead to increased plasmid stability in new hosts.

### **1.10 Computer Simulations**

The quantity of biological data has recently seen tremendous growth and now far surpasses the amount that any researcher can feasibly analyze. In light of such massive amounts of data, the discovery of large-scale patterns and correlations is proving to be essential. Computer models and simulations have subsequently become invaluable in the

biological sciences, assisting researchers in formulating hypotheses, analyzing large datasets, and answering important questions.

Recent technological advances have led to highly detailed models with many different components incorporated – for example, all bacterial cells in a population may be simulated, instead of a limited few. These models can provide quantitative insights where previously only qualitative descriptions were available, allow researchers to incorporate data from a variety of disciplines, and lead to an understanding of the interactions between various properties that were previously difficult to correlate.

Computational models have been used in many diverse areas of biological sciences, including the study of language (Cangelosi and Parisi, 2001), plant growth and development (Prusinkiewicz, 2004), and, recently, even a whole cell model with all of its molecular components and interactions (Karr *et al.*, 2012). Simulations have also been used to examine many aspects of plasmids (Bergstrom *et al.*, 2000; Levin *et al.*, 2000; Zhong *et al.*, 2012).

Though simulations have grown in popularity, there are some limitations on what a simulation can do. Because models are only representations of physical, chemical, and biological processes, they are, by definition, incomplete (Wooley and Lin, 2005). This is due to the fact that it would be incredibly difficult to model every biological component and because our understanding of most processes is incomplete. However, simulations are still extremely useful, especially in evolutionary studies. Simulations provide a faster way of analyzing what may happen over the course of many generations, which would take a long time in a laboratory setting.

### **1.11 Goals**

Antibiotic resistance is emerging as a worldwide problem, yet many of the evolutionary mechanisms underlying this resistance remain unknown. Plasmids in single-bacterial populations may undergo many mutations, including adapting to their hosts and further enhancing their sensitivity to antibiotics. Laboratory experiments attempting to understand

the rising diversity in a population are limited in the number of observable parameters and the possible correlations between these parameters. Generation time, combinations of parameters, and control conditions are all limitations in a laboratory setting that make it difficult to find any strong correlation between experiments. However, these are all easily testable by computer simulations.

Simulations can be used for an extended generation time, can test any combination of parameters, and do not have the need for control conditions found in a laboratory. And simulations can be run in a short amount of time, compared to laboratory experiments. Therefore, simulations can be used to make predictions and formulate hypotheses about plasmid evolutionary dynamics, which can later be tested empirically.

The work produced in this thesis revolved around the creation of a simulated model of plasmid evolution. The goal of the simulation is to assist in formulating hypotheses regarding factors that may play an important role in the evolutionary process of plasmid adaptation to its host, including how mutations in certain regions of the plasmid affect a plasmid's characteristics (including cost, loss rate, and conjugation rate) and population diversity. Plasmids in the simulation will be allowed to grow and conjugate with other hosts. Mutations that occur during growth and/or conjugation are noted and their effects on different parameters are implemented. Information on the relative frequencies of different mutations will assist researchers in determining which genetic regions and which plasmid characteristics have the biggest impact on population diversity.

## Chapter 2: Simulation

This computer model of plasmid evolution was developed in the R programming language (R Core Team, 2012). R is a language designed for use in statistical applications; it manages matrices efficiently and provides many built-in functions for performing statistical operations and data manipulation. RStudio (RStudio, 2012) was the chosen development environment as it provided a simple interface for writing and debugging code.

### 2.1 Input Parameters

The model takes a number of inputs into consideration before beginning the simulation. These parameters allow for many modifications to be made to the model to account for any hypothesis a researcher wants to test. The simulation then executes (taking into account the input) and the results can be used to help further formulate and develop hypotheses related to plasmid-host interactions.

Parameters used in the model include:

- Type of plasmids and hosts
- Population density
- Host doubling rate
- Plasmid mutation rate
- Plasmid cost
- Plasmid loss rate
- Plasmid conjugation rate
- Simulation length
- Bottlenecks
- Mutation regions and effects

***Type of plasmids and hosts.*** Different types of plasmids may have different attributes – varying rates for mutation, conjugation, and replication, different antibiotic resistances,

and belong to incompatibility groups. Hosts may also have different attributes including the initial stability and cost of the plasmids to the hosts (which will also be dependent on the level of selection) and the replication rate of the host. The type of plasmids and hosts is used in the code to help the researcher distinguish and designate different properties for each type. At the conclusion of the simulation, names given for the different types of hosts and plasmids will allow one to chart the progress of each type.

**Population density.** This parameter involves two values – one that indicates how many hosts will begin the population and another to dictate the maximum number of doublings the population can go through.

**Host doubling time.** This parameter will dictate how many iterations of the simulation will be needed in order for a generation to occur. If a host doubles 100% of the time, every iteration of the simulation will equal a generation. A lower doubling time, or growth rate, will require more iterations for each generation. This is important when determining when to bottleneck and is useful for comparing two different hosts with plasmids. This parameter is important when there are two hosts in the population that may have different growth rates. The simulation assumes a constant growth rate through the iterations.

**Mutation rate.** Different plasmids may have different mutation rates, which can also be dependent on the host the plasmid resides in. The mutational load for a plasmid is dependent on the size of the plasmid – mutation rates are typically per nucleotide, so a large base pair plasmid will likely have more mutations than a smaller one. In this simulation, the mutation rate is per plasmid, not nucleotide. A higher mutation rate may lead to more clonal interference (when different genotypes with unique mutations exist simultaneously in a population). However, a lower mutation rate may lead to zero mutations fixing in a population.

**Plasmid cost.** A plasmid encoding antibiotic resistance will cost more than a plasmid without genes for antibiotic resistance. The cost of the plasmid will slow down growth of the host. For instance, if host A has a doubling time of  $x$  and a plasmid with cost  $c_1$ , the cost of the plasmid will cause the growth rate to decrease as  $x*(1-c_1)$ . If the plasmid is



subsequently lost from the host, the growth rate will be  $x$ . Additionally, if a mutation decreased the cost of the plasmid, the growth rate would be higher,  $x*(1-c_2)$  where  $c_2 < c_1$ . However, in the presence of antibiotics, a host without an antibiotic resistance plasmid will likely be unable to replicate. Selective pressure is important in determining the cost of a plasmid as well as a plasmid's stability.

**Loss rate.** Different plasmids have different loss rates. This is important in determining how often a plasmid is likely to be lost through segregation when its host replicates. A host may replicate before its plasmid has replicated, or without proper plasmid partitioning to the cell poles, resulting in one daughter cell without a plasmid. This will lead to a number of outcomes, including the plasmid-free cell's doubling time being inhibited (in the presence of selection) or outcompeting the other cells in the population in the absence of selection, because it doesn't have the burden of the plasmid.

**Conjugation rate.** Different plasmids conjugate at different rates. The conjugation rate is important in controlling the amount of horizontal gene transfer occurring in a population. The probability of conjugation is dependent on the conjugation rate and number of recipients (hosts without a plasmid) in the population.

**Simulation length.** The length of the simulation is the number of iterations the simulation should run for. More iterations lead to more mutations and more time for lineages to sweep through or die out in the population. However, more iterations also take significantly more time and computational resources. One needs to figure out the optimal time to run the simulation for optimization of these parameters.

**Bottlenecks.** Many experiments run in a lab, in particular the serial batch cultures, will undergo multiple bottlenecks to help control the population size and to minimize the time the bacteria spend in the stationary growth phase. Bottlenecks will severely reduce the population size to a randomized set at specific points in time. For instance, every ten iterations, a population could be reduced to only a small percentage (i.e. a 1,000-fold reduction, as in Hughes *et al.* (2012)) of the current population size. This will help

determine which lineages control larger portions of the population and eliminate lineages comprising small portions of the population.

***Mutation regions and effects.*** It is important to note where in the plasmid a mutation is likely to occur. In some experiments, it may be necessary to determine what geno- and phenotypic changes will occur if more mutations happen in one plasmid region over another. In that case, a weight can be assigned to each region dictating the frequency of mutation there. Additionally, this could be useful if it is known that these mutation regions are different sizes. For instance, if the replication region of the plasmid is twice the size of the accessory region, it stands to reason that the replication region will have twice the number of mutations as the accessory region. For the simulations presented here, however, all regions were assumed to be equal in size and thus able to mutate at the same frequency.

There are four different regions in which a mutation can occur in a plasmid. These are the replication, stability, conjugation, and accessory regions. Mutations in these regions may affect the primary function of the region as well as the fitness of the host as a whole. The effect of the mutation will be determined by a uniform distribution with a mean of zero, where this value is used in calculating the change in a plasmid's cost, loss rate, and conjugation rate. Because the uniform distribution is centered around zero, a negative value is possible. These negative values are used to express a deleterious effect of the mutation. Any resulting negative probabilities are inverted. The same logic holds true for positive mutations as well. The distance of that result from zero will correlate to how much of an effect the mutation will produce.

Some mutations may be pleiotropic, meaning they may have an effect on multiple phenotypes (Lobo, 2008). For example, a mutation that increases the sensitivity of a plasmid's resistance to antibiotics will be extremely useful in the presence of antibiotics, but may also increase the cost of the plasmid. The increased cost of the plasmid will lower host fitness in the absence of antibiotics. Thus, mutations can be beneficial in some situations and deleterious in others.

*Replication region.* A mutation in the replication region may positively or negatively affect the plasmid's ability to replicate within a host, implemented in the code as its loss rate. If affected negatively, the probability of plasmid loss during segregation will increase.

However, if the mutation affects replication positively, it would reduce the chance of plasmid loss during host replication and segregation, resulting in an increased number of plasmids. A mutation in this region may also affect the cost of the plasmid. A plasmid with a more efficient replication mechanism would result in a lessened metabolic cost on its host. Therefore, in the simulation, mutations in the replication region of the plasmid will have an effect on both the cost and loss rate variables of the plasmid.

*Stability region.* A mutation in the stability region of the plasmid may positively or negatively affect the maintenance mechanisms that a plasmid may employ. If a mutation affects the stability region negatively, these mechanisms could falter and lead to a higher loss rate, resulting in an increase of plasmid-free cells. However, a beneficial mutation in this region would result in a lower loss rate of the plasmid and fewer plasmid-free segregants.

*Conjugation region.* A mutation in the conjugation region may affect the ability of the plasmid to transfer to other cells within the population and/or to other species of bacteria. Mutations in this region will have a direct positive or negative effect on the conjugation rate of the plasmid. A plasmid that is able to conjugate quickly and effectively will spread throughout a population much faster than others and have a better chance at survival. A positive mutation in this region within the model would enable a plasmid to spread through the population more rapidly. A negative mutation, however, would limit a plasmid's horizontal transfer capabilities and result in the plasmid primarily spreading through vertical transfer.

In biology, plasmid cost may be related to the horizontal spread of the plasmid but this isn't always the case and as such, in this simulation, these two attributes (cost and horizontal transmission) are not linked.

*Accessory region.* A mutation in the accessory region of a plasmid may affect a plasmid's resistance to harmful substances. Plasmids can confer antibiotic resistance and heavy metal resistance, among others, to their host. A beneficial mutation in this region could help to ameliorate the cost of these accessory genes or allow for a higher growth rate in the presence of selection (e.g. antibiotic or heavy metal) with no change in plasmid cost. A deleterious mutation, however, may disable the plasmid's resistance and return antibiotic sensitivity to its bacterial host. Additionally, a mutation in this region could have an effect on the cost of the plasmid. A mutation resulting in more accessory genes could lead to a higher cost plasmid. On the other hand, another mutation could also decrease the cost of these genes to the host.

Host fitness changes will be relative to the original, ancestral host. A host with a mutated plasmid may be slightly more or less fit than the ancestral host, due to a change in the cost or stability of the plasmid. These changes in fitness will affect the competitiveness of the host during replication.

## **2.2 Assumptions**

A few assumptions were made when creating this model. The first assumption was that the bacterial host does not mutate. This assumption was made because the project was focused on which plasmid mutations will affect a plasmid's host range. The project was also not aimed at determining coevolution between the host and plasmid. While this is artificial and does not reflect nature, this is a good starting point in developing a comprehensive simulation. The simulation could be easily expanded to include host mutations by adding a few more parameters and characteristics in the host class. However, care must be taken to ensure that host mutations are not confused with plasmid mutations when running the simulation.

A second assumption was that the effect weight of the mutations would be randomly decided, chosen by a uniform distribution. The effect weight of a mutation is how much of a change is caused by that mutation and may be either beneficial or deleterious. The

mutation will have an effect on different components of the plasmid, depending on its location. In one run, a mutation in the replication region may sweep through the population. However, in the next run, it may be possible for a mutation in the stability region to become more prevalent in the population. Additionally, all plasmid regions were able to mutate at the same frequency. Preference was not given to one region over another.

The final assumption was to exclude the vast majority of intracellular competition. In this simulation, conjugation only occurred between a donor host with a plasmid and a recipient host without a plasmid. Hosts could not have more than one plasmid. Additionally, stability mechanisms employed by a plasmid (such as copy number and toxin/antitoxin systems) were ignored in keeping with the removal of intracellular competition. Instead, plasmids are assigned a cost that is relative to what effect these mechanisms might have on the host.

### **2.3 Simulation**

This model simulates a typical experimental plasmid evolution study in serial liquid batch culture. Plasmids will replicate, conjugate, and mutate in ways that mimic experimental conditions. The goal of the simulation is to assist in formulating hypotheses regarding factors that may play an important role in the evolutionary process of plasmid adaptation to its host, including how mutations in certain regions of the plasmid affect the plasmid host range.

The simulation will follow the flow explained in Figure 4.

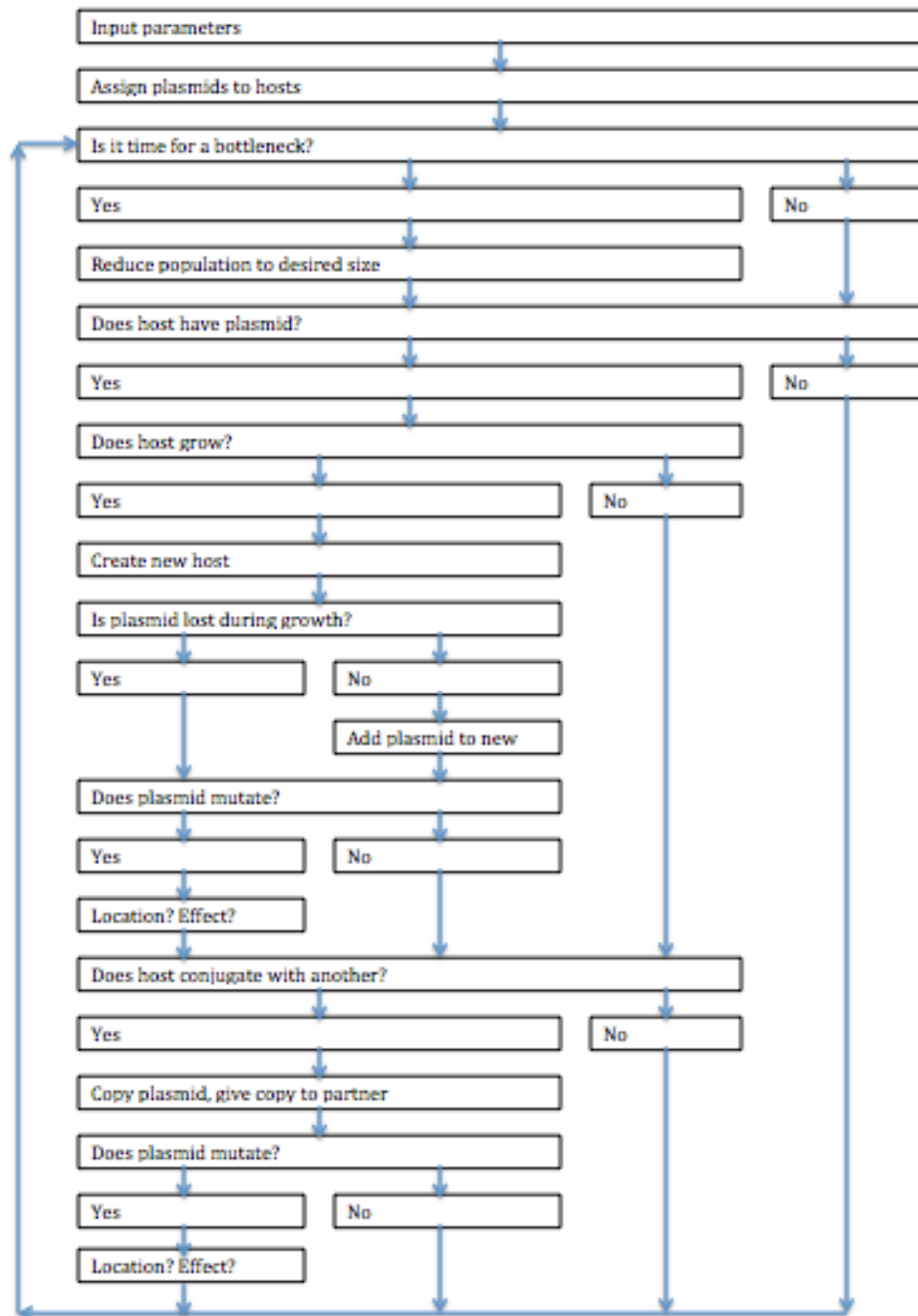


Figure 4: Steps of the simulation. After all plasmids and hosts are created, each host goes through the simulation to determine if it has a plasmid. If yes, it may grow and/or conjugate or do nothing. During growth and conjugation, a mutation may occur in the plasmid.

The simulation treats all hosts as individuals and continues in discrete time. It uses the input parameters described above to dictate the flow of the simulation. After all of the parameters are put into the program, the simulation starts by randomly assigning plasmids to hosts. If there are multiple hosts and/or multiple plasmids, a random distribution of all is recorded. If desired, it is also possible to allow some hosts to start the simulation without a plasmid. This case is only possible without antibiotics present and may be useful in determining the competitiveness of the ancestral host compared to that with a plasmid.

After all plasmids are assigned the simulation is fully initialized. It starts by determining if it is time for a bottleneck – a sharp reduction in the size of the population as a result of random events. This happens at the beginning of each cycle of the simulation. If a bottleneck occurs, the host population is reduced to the desired size (a percentage of the population) before continuing as normal. The bottleneck chooses a random assortment of hosts from the current population to continue through the simulation.

The simulation then runs through each of the hosts individually to determine if they contain a plasmid. If this is true, then the host may conjugate and/or divide, or do nothing. The probability of the host conjugating depends on the plasmid's rate of conjugation and the number of recipient hosts (number of hosts without a plasmid) in the population. The probability of division depends on the doubling rate and the cost of the plasmid.

$$\textit{Probability (conjugation)} = \textit{Plasmid conjugation rate} * \textit{Number of recipient cells}$$

$$\textit{Probability (growth)} = \textit{Host growth rate} * (1 - \textit{Plasmid cost})$$

If a conjugation occurs, two plasmids may reside in the same cell. If these plasmids are of the same incompatibility group, only one survives – the other is lost (chosen randomly in this simulation). However, if the plasmids are of different incompatibility groups, they will be able to reside in the same cell, at a greater cost to the host. In general for the simulation runs described in this work, however, conjugation occurred only between a donor host with a plasmid and a recipient without a plasmid.

If division occurs, the single cell (plasmid host) divides into two identical bacteria with a copy of the plasmid in each. However, it is possible that the host's plasmid may be lost during segregation. The loss rate of the plasmid comes into play here. A plasmid may be lost in one of the daughter cells when a host divides during replication if the plasmid does not equally segregate to each side.

In either case – conjugation and division – the simulation continues to determine if a mutation happens, based on the mutation rate set by the user (see Mutation rate in section 2.1 and Tables 1 and 2). If a mutation occurs, the location of the mutation is randomly determined – whether it is in the replication, stability, conjugation, or accessory region of the plasmid. The effect (beneficial or deleterious) is next determined by the uniform distribution and any necessary changes are recorded. Using the uniform distribution, centered on zero, the effects can be positive or negative, where negative values are used to express a deleterious effect of the mutation. The user may set the range of values for this distribution. The magnitude of the effect determines the overall impact of the mutation. For example, a mutation with an effect of 5 in the stability region of the plasmid will result in a loss rate that is five times lower than the ancestral loss rate. Mutations may affect the plasmid's loss rate, conjugation rate, incompatibility, accessory genes, and/or cost, dependent on the location of the mutation.

If a host does not contain a plasmid, it may divide, but will not conjugate or mutate. The host without plasmid also can only divide when there is no antibiotic selection for the plasmid. It will remain in the state it is in until such a time when it either conjugates with another host and obtains a plasmid or is removed from the population through a bottleneck. There is no cell death in this simulation, as it models a serial batch culture, where cells spend only a limited amount of time in the stationary phase and thus little to no cell death occurs.

Each type of host will then be counted, including the number of plasmids and the number and location of the plasmid's mutation(s). These will be counted after each time point (iteration) in order to observe the changes in the lineages over time. The simulation will



then begin again, starting at determining if a bottleneck will occur this iteration. These counts will be saved, enabling the researcher to compare many aspects of the simulation.

The entire simulation can be repeated many times with the same parameters by indicating that multiple lineages are being tested. This will help to provide the researcher with an example of many different possible outcomes for the same simulation.

## **2.4 Output**

For each run in each lineage, there is a file that explicitly details all of the hosts and plasmids included in the simulation at this time. It details host number and type; from which parent host this divided from and which iteration this division occurred; if it is viable or not and, if not, during which iteration it became non-viable; how many times it has divided and its doubling rate; how many plasmids it has; the type of plasmid(s) it has and all of the information (attribute, incompatibility, cost, loss rate, mutation rate, and conjugation rate) connected to that plasmid; any mutations the plasmid incurred, along with its type, fitness, location, and effect; and, if this host conjugated with another, the conjugation partner and during which iteration conjugation occurred. An example of this output can be seen in Figure 5 below.

These results are collected using a script written in Python. Python was used because of its ability to parse strings simply and efficiently. The text information is read in and converted to data. This keeps track of the number of hosts without plasmids, the number of hosts with non-mutated plasmids, and every mutation or sequence of mutations that a plasmid may have. Every different mutation or combination of mutations is treated as an individual – if a secondary mutation occurs, it is labeled as a different type and not as offspring of the first. In this way, it is easy to determine how many different genotypes occurred in the simulation and the number of hosts and plasmids that have that genotype. Additionally, it's easy to identify which mutations became prominent in the population and which died out relatively quickly. This information is written to a Microsoft Excel file, where the data may be analyzed. In some of the following simulations, the number of different genotypes was

too large for Excel to handle. For this reason and to maintain consistency throughout, only the top 250 most common genotypes for each simulation were written to this file for analysis. Plasmids with zero mutations and hosts with zero plasmids were considered genotypes and were therefore subject to this limitation, as well.

```
Host: 15114 , A
  Parent: 14527, Iteration: 48
  Num Divisions: 14, Replication Rate: 1
  Plasmid (1):
  Type: P, Attribute: Antibiotic Resistance,
  Incompatibility: P, Cost: 0.02317412
  Loss Rate: 0.002, Mutation Rate: 0.005, Conjugation
  Rate: 6.472737e-06
    My Mutations (1):
    Location: Conjugation, Effect: 6.472737
    Conjugation: NA, Iteration: NA
```

Figure 5 Example of the information kept for each host during each repetition of the code.

## 2.5 Running the Simulation

Each run was conducted using the University of Idaho's Institute for Bioinformatics and Evolutionary Studies (IBEST) Computational Resources Core (CRC). The Core is a cluster of computers, with 512 processors and 2 terabytes of memory. For comparison, the average desktop computer has 2 processors and 8 gigabytes of memory. Because of the distributive nature of the cluster, multiple runs could be submitted simultaneously, without loss of processing power or memory. The run time for each simulation depended on the number of hosts, lineages, and repetitions; most simulations were completed in a matter of hours, though some took a few days.

## Chapter 3: Results

This simulation was created to assist in formulating hypotheses about how mutations in certain regions of the plasmid affect the evolutionary dynamics of a plasmid. First, multiple simulation runs were performed, varying a combination of parameters to observe how each parameter affected the outcome of the simulation. Second, to validate the simulation, simulations were run to try to mimic published plasmid evolution experiments to observe the similarities between the experimentally determined results and the simulated results.

### 3.1 Simulation Parameters and Values

There were many different parameters involved in the model, which can be viewed in Table 1. These are all parameters that a user may change in order to simulate different experiments. Included in Table 1 are the corresponding variable name for the parameter in the code, the value used in the following results, and a short description. A longer description can be found in Section 2.1 of this thesis.

**Table 1 Simulation parameters and values**

<b>Parameters</b>	<b>Variable name</b>	<b>Value</b>	<b>Description</b>
Type of hosts and plasmids	host_type, plasmid_type	Host "A" and Plasmid "P"	Name of one or more hosts or plasmids
Population density	num_hosts, pop_limit	$10^3$ and $10^5$	Initial and maximum population sizes
Host doubling rate	growth_rate	1 per iteration	Important if there are multiple hosts with different generation times
<i>Plasmid mutation rate</i>	mut_rate	Variable	Mutation rate per plasmid
<i>Plasmid cost</i>	cost	Variable	Fitness cost a plasmid imposes on its host
<i>Plasmid loss rate</i>	loss_rate	Variable	Rate of segregational loss of a plasmid
<i>Plasmid conjugation rate</i>	conj_rate	Variable	Transfer frequency of a plasmid
Plasmid attribute	attribute	Antibiotic resistance	Dictate if a plasmid confers any benefit to the host, i.e. antibiotic resistance
Simulation length	rep	250	How long to run the simulation for (number of iterations)
Bottlenecks	bottleneck_gen, bottleneck_pop	7 and $10^3$	Bottleneck after a specified number of generations, to a specified size
Mutation regions	mut_loc_prob	0.25, 0.25, 0.25, 0.25	All plasmid regions are equally likely to have a mutation
Mutational effect	effect	Range from -10 to 10 times the ancestral value	The range of effects a mutation may have on a plasmid phenotype

For the following simulations, only one bacterial host type and one plasmid type were used. This was to examine the plasmid diversity that can arise even in a single population. The plasmid was assigned an antibiotic resistance attribute and the simulation was run under antibiotic pressure, where hosts without a plasmid were unable to replicate. The initial population size was  $10^3$  and the maximum was  $10^5$ . In order to finish the simulations in a reasonable amount of time (days rather than weeks) the population size was limited to a maximum of  $10^5$ . However, this limit could be increased, with the only major cost being time. These values were used to encompass a large population while still keeping a host replication rate that was similar to laboratory rates. The host, on average, doubled during every iteration of the code – keeping one iteration of the model equivalent to one

generation. A bottleneck was incorporated every 7 generations and reduced the population size down to  $10^3$ . Additionally, mutations were equally likely to occur in any of the four mutation regions. The simulation was run for 250 generations.

The four parameters that are italicized in Table 1 were used as initial ancestral values in the simulations that generated the results described below. The values – high, medium, or low – were taken from various published studies. The high and low were both rather extreme, whereas the medium value was a fairly general representation. The low values for cost and conjugation rate were set to zero because it is easy to conceive of a plasmid that imposes no cost to its host and one that is not self-transferable. Every combination of parameters at high, medium, and low values were used, for a total of 81 combinations.

These values were chosen for the effect a mutation might have on them, their impact on a plasmid's stability, and their ability to influence the level of diversity in a population. The cost of a plasmid affects the stability of the plasmid and its host's ability to grow (and thus increase its population size), the conjugation and loss rates of the plasmid are directly correlated to the stability of the plasmid in the population, and the mutation rate dictates the level of diversity in the population. Additionally, these values are all related – a low cost plasmid will allow for more growth and, as a result, more mutations are likely to happen. Similarly, a high loss rate will produce more segregants, or hosts without a plasmid, in the population, which in turn has an effect on conjugation – more recipient cells in the population leads to a higher probability of conjugation. And again, mutations may happen during conjugation, further diversifying the population.

**Table 2 Variable parameters and their values**

Parameter	High	Medium	Low
Cost	0.5 <sup>1</sup>	0.05 <sup>1</sup>	0
Mutation Rate	0.5 <sup>2</sup>	0.0005 <sup>2</sup>	0.000005
Conjugation Rate	0.1 <sup>3</sup>	0.0001 <sup>4</sup>	0
Loss Rate	0.5 <sup>5</sup>	0.005 <sup>6</sup>	0.00005 <sup>1</sup>

<sup>1</sup>De Gelder, et al, 2007. <sup>2</sup>De Gelder *et al.*, 2004. <sup>3</sup>Klaenhammer and Sanozky, 1985. <sup>4</sup>Tomita *et al.*, 2003. <sup>5</sup>Ansari and Gartenberg, 1997. <sup>6</sup>Boe and Rasmussen, 1996.

### 3.2 Running the Simulation

The simulation was run 81 times, with every combination of the plasmid cost, mutation rate, loss rate, and conjugation rates changing to high, medium, or low as the initial ancestral value.

Graphs were created from the simulation output in order to visually compare the population dynamic differences. The graphs indicate what percentage of the population has a plasmid or not and if that plasmid mutated once or multiple times. The genetic region wherein the mutation(s) occurred is shown in the graph, along with what percentage of the population underwent that particular mutation. In all graphs, colors are used to represent the initial mutation in different genetic regions. The stability region is represented by the color green, the replication region is orange, the conjugation region is blue, and the accessory region is purple. Ancestral plasmids, thus those without any mutations, are represented by the color red and hosts without any plasmids are yellow. Differing hues of the same color represent different genotypes. For instance, different hues of blue indicate multiple mutations stemming from an initial mutation in the conjugation region. Some plasmids have mutations in different regions, for example with the first in the conjugation region and the second in the replication region, or even three mutations, in the conjugation, accessory, and, again, conjugation regions. These are all colored in hues of blue in the figures. Each different colored section represents a different genotype. This graphical output was to help make the graphs easier to read and correlate. Additionally, it is always possible to return to the original output files and see the details of the actual mutants that arise to determine whether or not they rise in frequency and eventually sweep or not.

Different combinations of parameters will result in vastly different populations. In Figure 6, it is possible to see this. Figure 6a shows the combination where every parameter was set to Low. While there was some diversification, it did not happen often. Conversely, Figure 6b shows every parameter set to Medium. This allowed for improvement of every parameter, though mutations in this run can mainly be seen in the replication (orange) and

accessory (purple) regions. Mutations in these two regions had an effect on the cost of the plasmid, with the replication region also having an effect on the plasmid's loss rate. A lower-cost plasmid allowed its host to divide more frequently than hosts with ancestral plasmids and, thus, spread through the population more rapidly.

In Figure 6b, it's also possible to see a high amount of clonal interference in the population. Clonal interference is the competition between different genotypes with various beneficial mutations (de Visser and Rozen, 2006). Mutations in both the replication and accessory regions of the plasmid lowered the cost of the plasmid and both genotypes are competing in the population. This is prevalent even in the reduced population size in the simulation.

From Figures 6a and 6b, it is possible to interpret that the differences in the graphs are due solely to the changes in the mutation rate. However, this may not necessarily be the sole cause of the differences. Additionally, it is also important to note the effects from the other three parameters. Therefore, in order to determine what effects each of the four studied parameters had on the overall population diversity, each of these parameters was studied individually (which can be seen in Sections 3.3-6).

Note that the simulation includes a certain amount of innate randomness. Randomness occurs in the location of the mutation, the size of the effect of the mutation, and the bottlenecks. The simulation, run multiple times with the same parameters, may produce varying results. Future work could involve modeling the probabilities of mutations in various regions. An assumption of the model was to allow all plasmid regions to mutate at the same frequency.

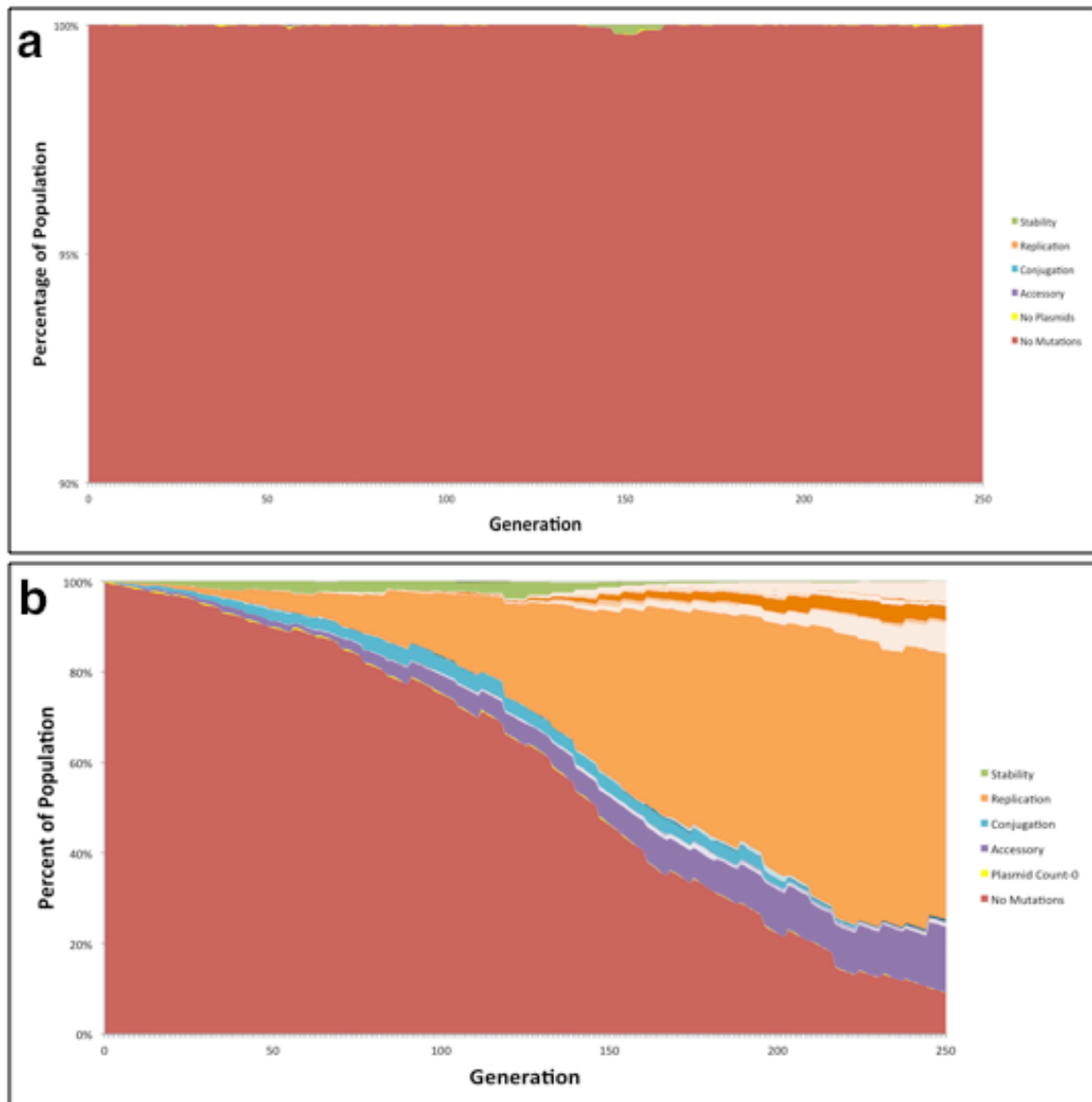


Figure 6 Differences in evolutionary population dynamics due to differences in parameters of the ancestral plasmid. Mutations in the stability region are shown in green, replication region mutations are in orange, conjugation region mutations are shown in blue, and accessory region mutations are purple. Hosts with no plasmids are shown in yellow and plasmids with no mutations (i.e. the ancestral plasmids) are in red. (a) Low cost, low mutation rate, low conjugation rate, and low loss rate. (b) Medium cost, medium mutation rate, medium conjugation rate, and medium loss rate.

### 3.3 Mutation Rate

The plasmid mutation rate has a large impact on the diversity of the evolving population. Indeed, population diversity was characterized almost solely by the number of mutations that occurred during the simulation. When the mutation rate was low ( $5 \cdot 10^{-6}$ ), as in Figure 7a, it took quite a number of generations before the population was able to diversify much.



However, an extremely high mutation rate ( $5 \cdot 10^{-1}$ ), as in Figure 7b, led to an incredible amount of diversity that left no room for a mutation to sweep through the population. Additionally, even the top 250 mutations in the population did not last very long and the resulting graph is difficult to read. There were large gaps in the graphs due to the sheer number of different mutations that occurred – many genotypes had very few constituents and, therefore, it was difficult to discern the amount of variance in later generations. As a result of this, Figure 7b only shows the first 28 generations. After this, the population varied too much to be discerned. However, it is still possible to observe the incredible amount of population diversity that quickly arose in the population.

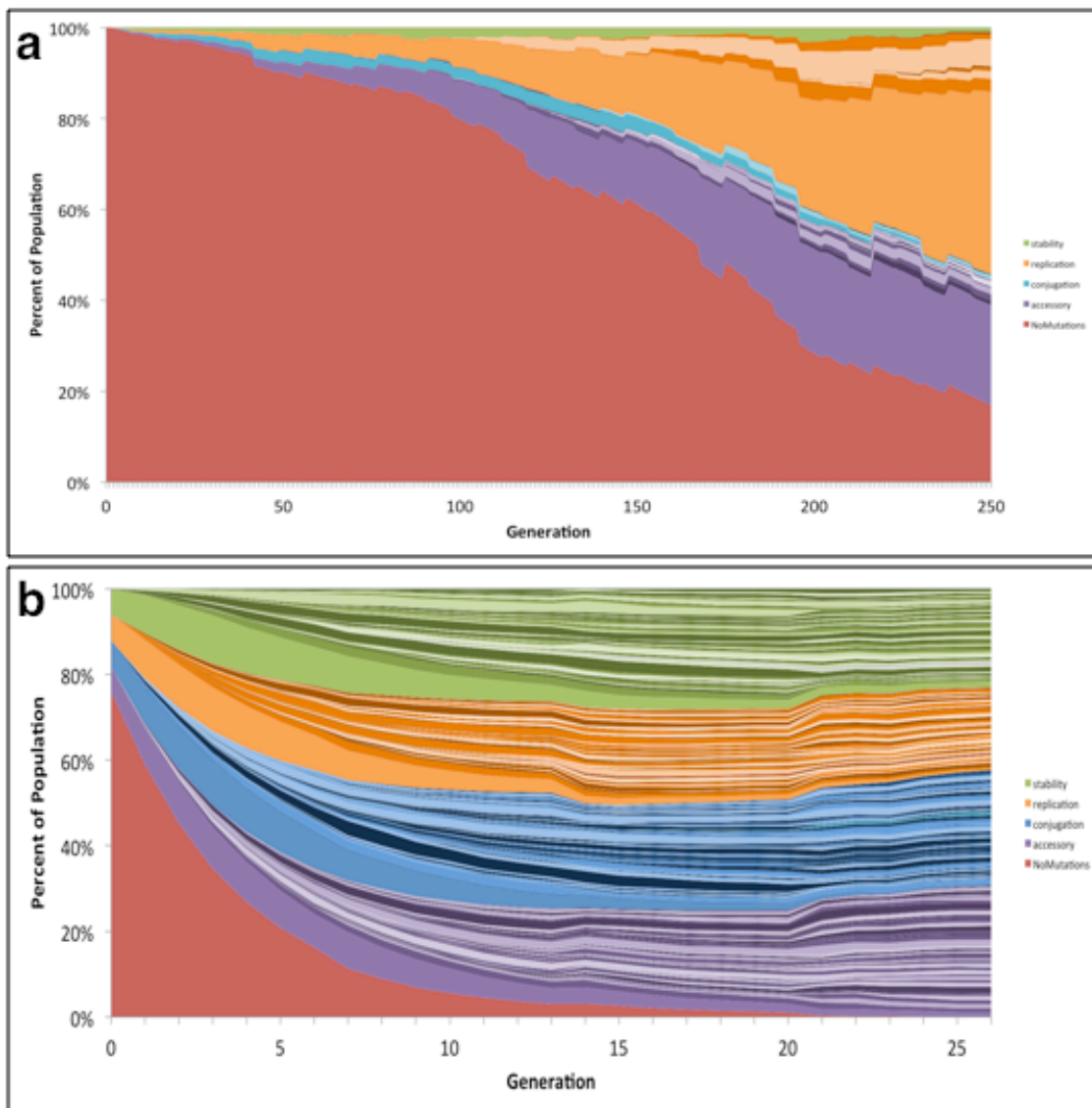


Figure 7: Comparison of population evolutionary dynamics for ancestors with different mutation rates. With cost, conjugation rate, and loss rate set to the medium value, the mutation rate was (a) low and (b) high. Green indicates mutation in the stability region, orange for the replication region, blue for the conjugation region, and purple for the accessory region. Red shows no mutations, or the ancestral plasmid.

### 3.4 Conjugation Rate

Having a high conjugation rate may be extremely beneficial for a plasmid. Horizontal transfer will allow the plasmid to spread throughout the population and reduce the number of plasmid-free segregants. A beneficial mutation in the conjugation region of a plasmid will increase the frequency of horizontal transfer, and thus lead to an increased proportion of that plasmid's mutant genotype in the population. Additionally, as plasmid-

free segregants are not allowed to grow in the simulation, hosts that are re-infected with plasmids are again able to grow, further spreading its genotype in the population. Figure 8 depicts the differences in output between simulations starting with different conjugation rates. The graph in Figure 8a depicts the results of an experiment where the plasmids in the population initially had a low conjugation frequency ( $1 \cdot 10^{-7}$ ). However, when a mutation occurred that increased this frequency, it was able to quickly become prominent in the population. This is seen in the large blue section in Figure 8a. In Figure 8b, the plasmids had an initially high conjugation frequency ( $1 \cdot 10^{-1}$ ) and there wasn't much room for an increase. An increase here didn't have quite the sweeping effect as it did in Figure 8a. Instead, other mutants were able to successfully compete in the population (see larger regions of orange and purple in Figure 8b).

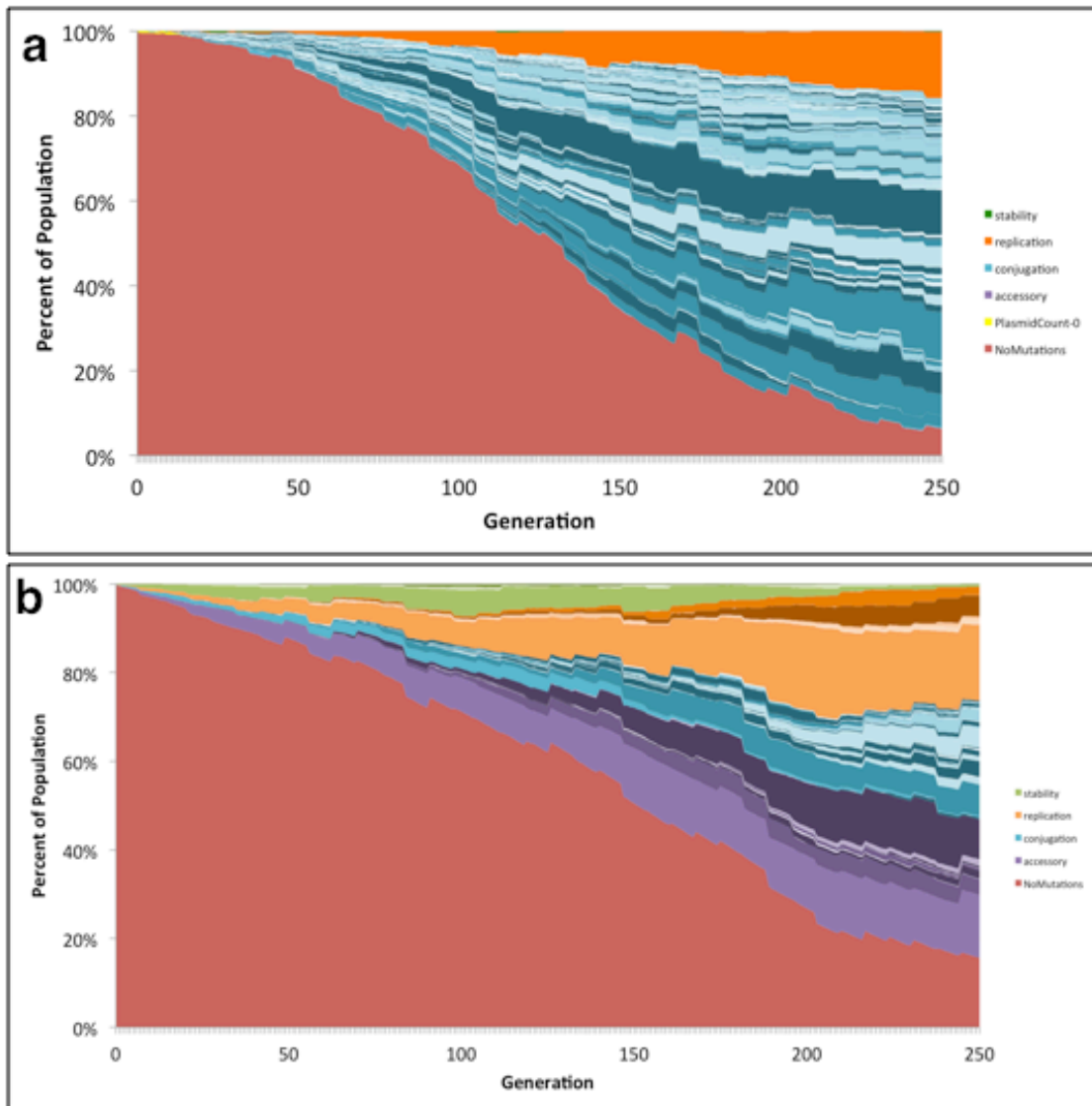


Figure 8: Comparison of population evolutionary dynamics for ancestors with different conjugation rates. With cost, mutation rate, and loss rate set to the medium values, the conjugation rate was (a) low and (b) high. Green indicates mutation in the stability region, orange for the replication region, blue for the conjugation region, and purple for the accessory region. Red shows no mutations, or the ancestral plasmid. When ancestral conjugation frequency was low (a), plasmids with mutations to increase the conjugation frequency (seen in blue) out-competed most other genotypes. However, when the frequency was high (b), other beneficial mutant genotypes (orange and purple) were able to survive in the population.

### 3.5 Loss Rate

A low plasmid loss rate means a high probability that a plasmid will be inherited by both daughter cells upon cell doubling. In contrast, a high loss rate leads to many plasmid-free segregants in the population. Thus, the loss rate is also very connected to the conjugation rate. More plasmid-free segregants in the population allows for more conjugation to occur,

as conjugation in this simulation only occurs between a donor host with a plasmid and a recipient host without one. Figure 9 shows the differences in output between simulations starting with different loss rates. When the loss rate is low, as in Figure 9a, not much diversity occurs because (i) there isn't much room for improvement of the loss rate and (ii) with few plasmid-free segregants, not much conjugation may occur. Instead, there was room for other parameters to improve. Mutations in the accessory region allowed for a lower-cost plasmid, which was beginning to out-compete the ancestral plasmid towards the end of the simulation. In Figure 9b, the initial loss rate is very high and, initially, a number of plasmid-free segregants are formed. A beneficial mutation in the replication region would have led to a lower loss rate. However, an interesting thing to observe is how conjugation is affected by the high loss rate. In this run of the simulation, a mutation allowing for more conjugation was favored over one that lowered the loss rate, possibly due to the assumption that plasmid-free cells are unable to grow. With a high loss rate, conjugation occurs frequently and those plasmids with a high conjugation rate are able to spread throughout the population very quickly.

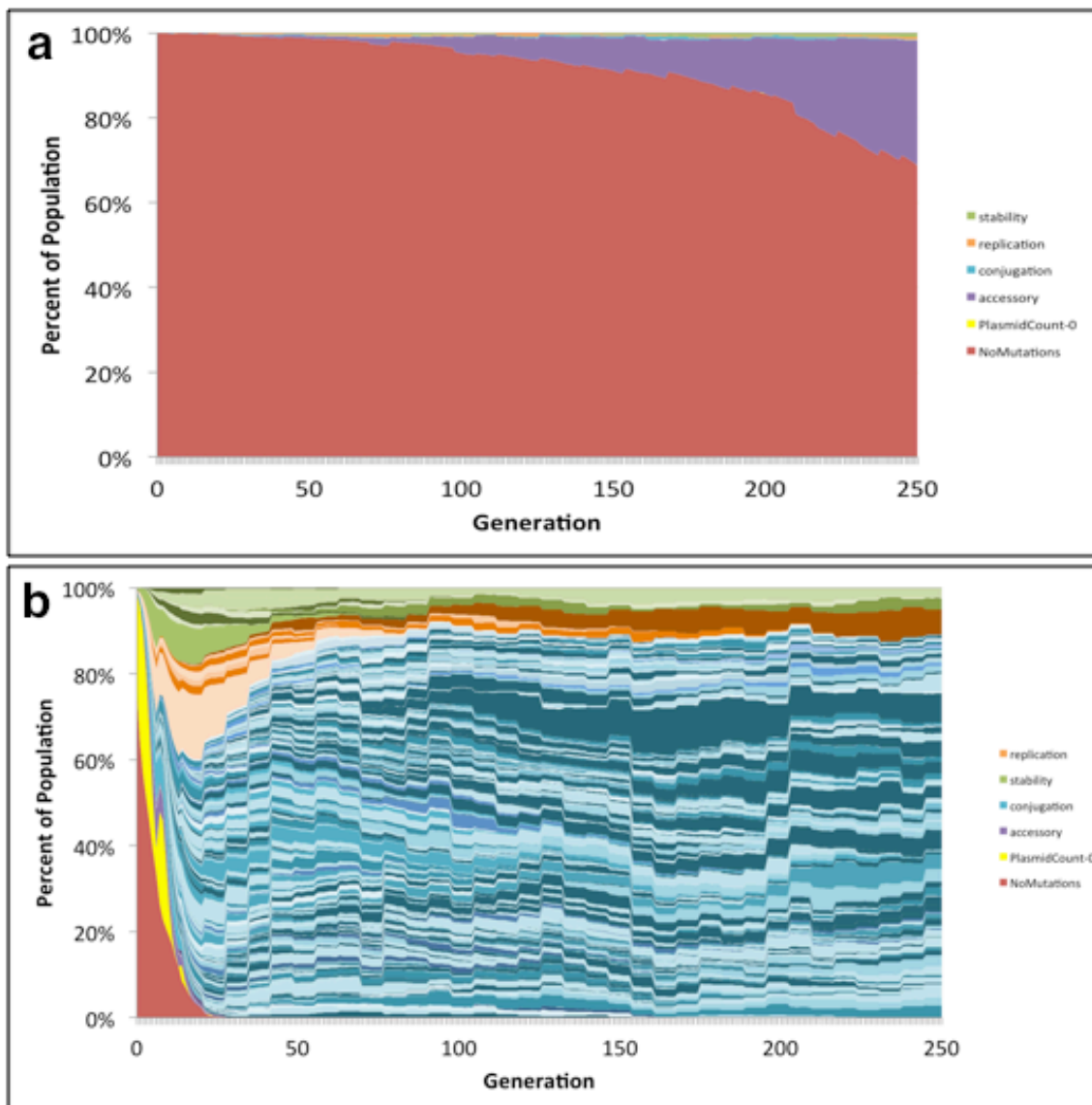


Figure 9: Comparison of population evolutionary dynamics for ancestors with different loss rates. With cost, mutation rate, and conjugation rate set to the medium values, the loss rate was (a) low and (b) high. Green indicates mutation in the stability region, orange for the replication region, blue for the conjugation region, and purple for the accessory region. Red shows no mutations, or the ancestral plasmid. When the ancestral loss rate was low (a), mutations didn't occur as frequently and weren't as successful as when the ancestral loss rate was high (b). A high loss rate allowed much room for improvement.

### 3.6 Cost

Plasmid cost impedes the growth of the host and, a slow-growing host will have difficulty competing and securing a place in the population. Therefore a plasmid imposing a lower cost to its host is more ideal than one with a high cost. Figure 10 shows the differences in simulated population diversity due to differences in initial plasmid cost. Figure 10a shows

the dynamics for an ancestor with a low-cost plasmid. As in the previous section (3.4 Loss Rate), a low-cost plasmid leaves little room for improvement and so it is difficult for a mutation to sweep through the population. Figure 10b shows a simulation where the initial plasmid cost was very high. Beneficial mutations in the replication and accessory regions, which lowered the cost of the plasmid, were quickly able to sweep and fix in the population. As a result of the lowered cost, hosts with these mutated plasmids were able to grow much more quickly compared to those with the ancestral plasmid.

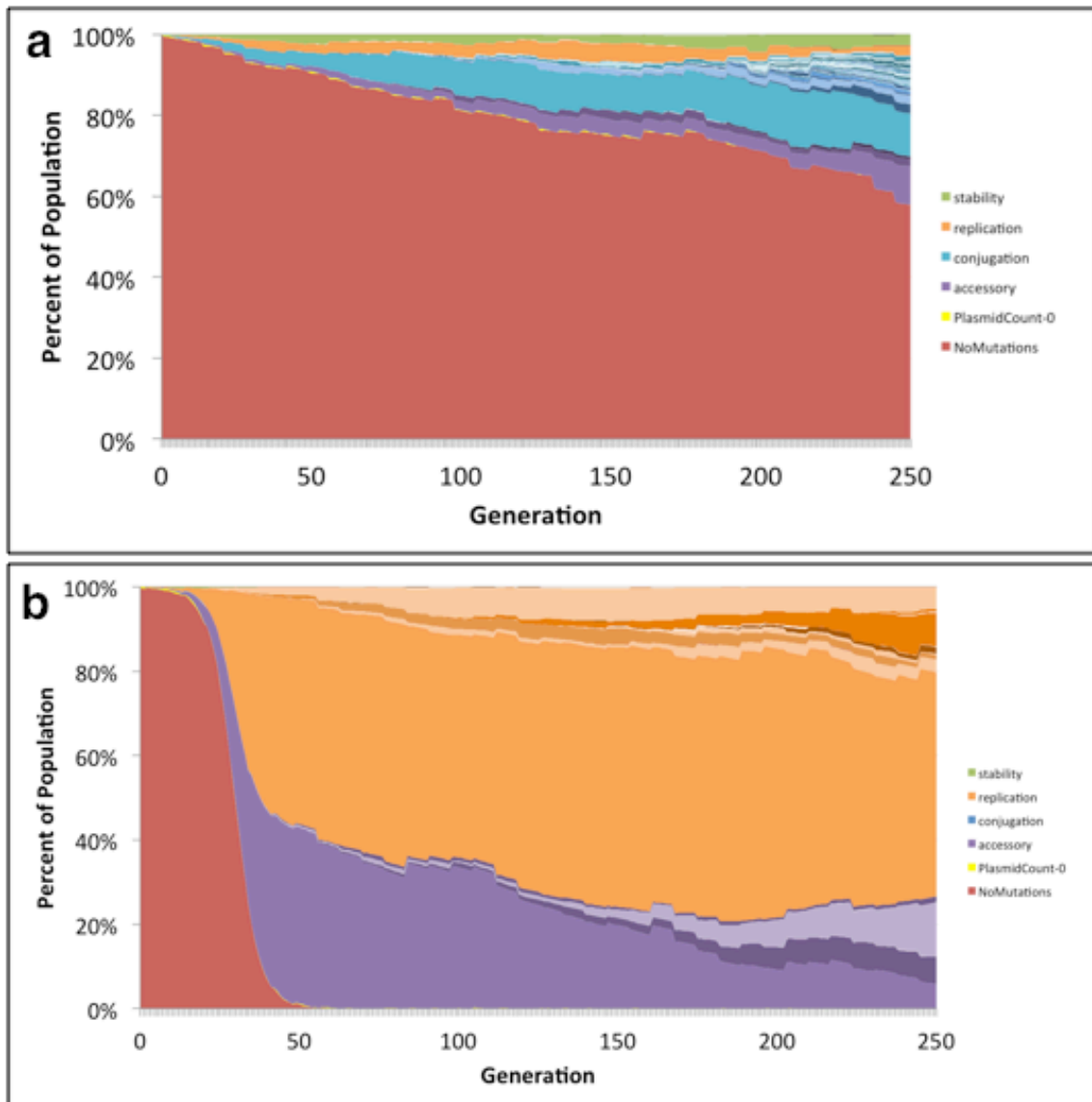


Figure 10: Comparison of population evolutionary dynamics for ancestors with different costs. With mutation rate, conjugation rate, and loss rate set to the medium values, the cost was (a) low and (b) high. Green indicates mutation in the stability region, orange for the replication region, blue for the conjugation region, and purple for the accessory region. Red shows no mutations, or the ancestral plasmid. With an initial high cost (b), mutations in the replication and accessory regions (lowering the cost of the plasmid) rapidly spread through the population.

### 3.7 Principal Component Analysis

Six statistics were gathered on all 81 simulations to help explain and clarify the observed dynamics and diversity. These are detailed below.

1. *Total number of genotypes.* The first statistic was the total number of genotypes that arose in the population at any time point during the 250 generations. As stated previously



(see 2.4 Output), only the top 250 genotypes were analyzed for each run. Therefore, the maximum number for this statistic was 250. The minimum value for this statistic was one and only occurred if zero mutations happened and no plasmids were lost from the population. This was informative as it helped to show the number of mutations that occurred during the simulation and the number of different possible genotypes.

2. *Richness*. Another statistic was the number of final genotypes, or richness – that is, the number of different genotypes present at generation 250, also with a maximum value 250. The minimum value for this statistic was zero and occurred when zero of the top 250 genotypes survived until the end of the simulation. This statistic was useful in observing species richness.

3. *Population evenness*. The third statistic was the population evenness at generation 250. Evenness is a measurement used when describing the diversity of a population. An uneven population is one in which one genotype dominates the population while the rest have very few constituents. This may be the product of a mutation sweeping through the population. Additionally, this could be caused by very few mutations occurring during the simulation, resulting in many ancestral plasmids remaining at the end. An even population is one that contains roughly an equal number of constituents per genotype in each generation. An even population depicts clonal interference – where different genotypes with beneficial mutations are competing in a population.

4. *Generation until half the plasmids have mutated*. A fourth statistic used was the number of generations until half of the plasmids in the population had at least one mutation. This was important in understanding how quickly mutations appeared in the population and could provide information about mutations spreading through the population.

5. *Generation until all the plasmids have mutated*. The fifth was the number of generations until all of the plasmids in the population had mutated (i.e., the number of generations until not a single ancestral plasmid was left in the population). This was useful in observing the correlation between the occurrence of mutations and the overall number of genotypes and final evenness in the population. Mutations occurring in later generations generally led

to a less even population, as there was not much time for those mutants to spread in the population. Mutations occurring earlier in time had a longer chance to spread through the population, but also allowed for much more diversity in the population.

These two statistics involving the number of generations until mutations occurred were separated into groups – 0, if the mutations happened in the first 50 generations, 1 for 51-100 generations, 2 for 101-150, 3 for 151-200, 4 for 201-250, and 5 for over 250 generations. This grouping was done for ease of comparison. Populations which took over 250 generations for half or all of the plasmids to mutate needed to be compared to those which took less time. However, this was impossible to do when simply using the generational number, as this number was unknown. Therefore, the grouping method was used to be able to compare these generational times.

6. *Generations all hosts have a plasmid.* This grouping method was also used for the last statistic, which was the total number of generations where all hosts in the population contained a plasmid. This statistic was useful in observing the effect of the conjugation and loss rates on the dynamics of the population.

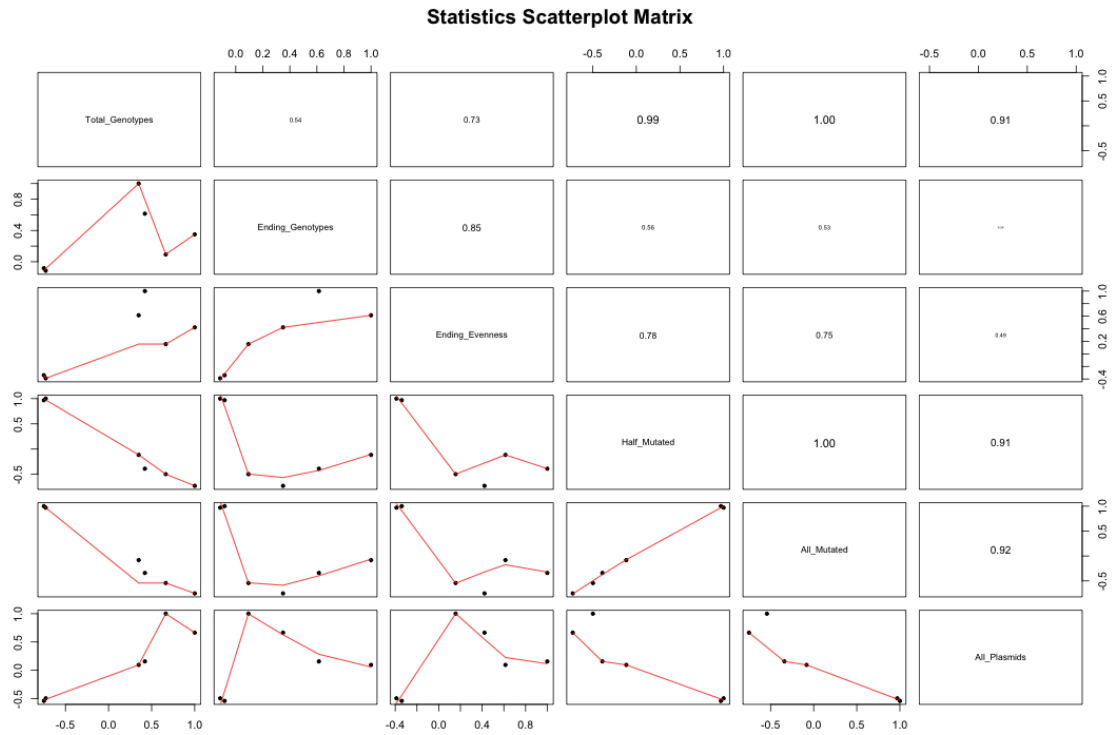
With these statistics, a principal component analysis (PCA) was performed. PCAs are used to find low-dimensional representations of the variation in a multivariate data set and provide a way of seeing underlying patterns in the data. This is done by creating new variables, called principal components (PCs), which are linear combinations of some or all of the original parameters. First, all of the variables were standardized. Standardization is a technique used in statistics to transform all variables into unit-less numbers. This allows for a comparison between statistics with different units and variances. Standardization was necessary because of the differences in the types of values needed for the statistics. In this way, the small range (between 0 and 1) of the evenness could be compared to the large range (between 1 and 250) of the total number of genotypes. After standardization, the PCA was run. Table 3 shows the proportion of variance for all PCs. The first principal component, PC1, always contains the most variation, with the following written in order of

decreasing proportion of variance. For example, PC1 explains 56.5% of the variance in the simulations.

**Table 3 Proportion of variance for all principal components**

	PC1	PC2	PC3	PC4	PC5	PC6
Proportion of variance	0.56543	0.23152	0.11026	0.05605	0.03213	0.00460

It is important to know the correlation between the statistics used in the PCA, as highly correlated statistics are redundant and can be reduced. Figure 11 shows scatterplots of these correlations. As expected, the generations when half of the plasmids in the population have mutated and when all of the plasmids have mutated are highly correlated. This means that these statistics were redundant and did not necessarily have to both be included in the PCA.

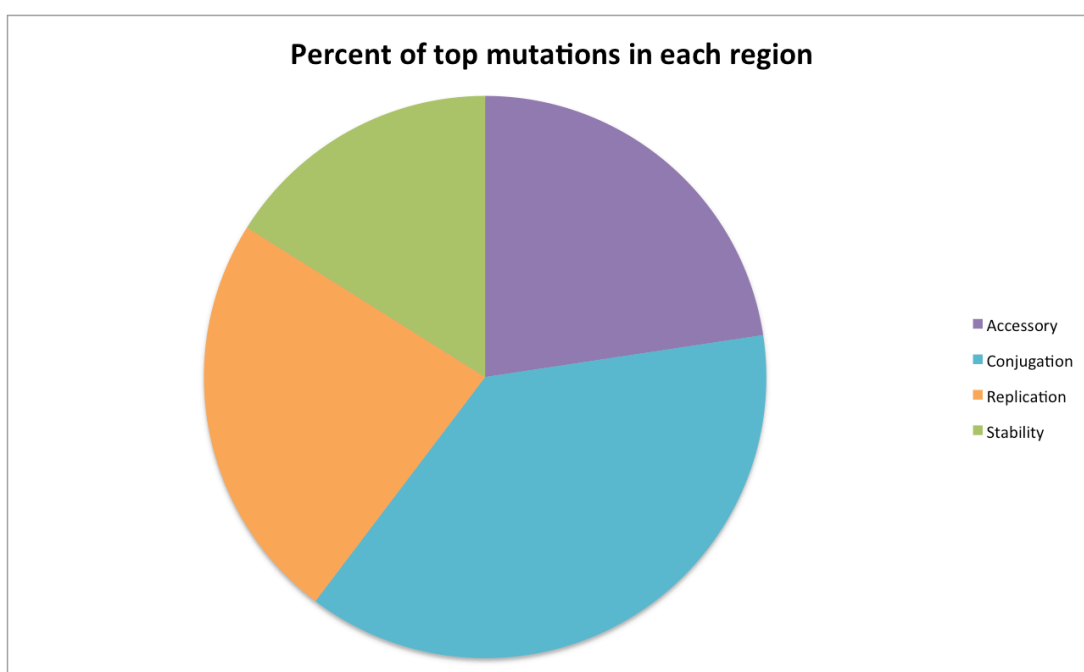


**Figure 11: Scatterplot of correlation of statistics. The graphs in the bottom left show a graphical comparison while the upper right contains a numeric description of the correlation. The size of the number is relative to the correlation, as well – a larger number size means a higher correlation.**

In Figure 12, the first two PCs are plotted against each other. The first two principal components show where the most variation occurs in the data. Next to each data point is an indicator of which values were used for each of the four parameters. The values are always in the same order – the plasmid cost is first, mutation rate second, conjugation rate third, and the loss rate is labeled last. This was used to identify patterns in the data. For example, the majority of the points on the far right of the graph indicate simulations where very little conjugation occurred (having lower conjugation and loss rates). Conversely, the majority of the points on the left of the graph indicate simulations where large amounts of conjugation occurred. This indicates distinct patterns in the simulations with low or high conjugation frequencies, essentially regardless of the values for the cost or mutation rate.



Another informative statistic was the frequency of mutations in different plasmid regions. When starting the simulation, all regions were equally likely to mutate. A record was kept of how many mutations stemmed from an initial mutation in each region – meaning, where the initial mutation in the plasmid occurred that provided such a benefit that this mutant genotype was able to survive in the population. This can be seen in Figure 13. This figure was summed over all 81 simulations and includes an analysis of over 11,000 mutations. It is interesting to observe that in a majority of cases, most of the first mutations were in the conjugation region of the plasmid.



**Figure 13: Percent of total mutations stemming from each mutation region for all simulations. Most mutations that swept through the population resulted from an initial mutation in the conjugation region.**

### 3.8 Validation

In order to validate the simulation, parameters were chosen that matched an experimental evolution study previously performed. When validating the experiment, it was expected that the results from the simulation would be similar to the results in the publications, though likely not match exactly. This is due to the simulation not fully encompassing every aspect of biology. However, the trends between the two should match fairly well.

Two publications were chosen and certain experiments from those papers were replicated. The first chosen experiment was from De Gelder *et al.* (2007), using the 64.5 kb plasmid pB10 in the host bacteria *Stenotrophomonas maltophilia* (P21), *Pseudomonas putida* (H2), and *Pseudomonas koreensis* (R28). This study determined the stability of pB10 in all three hosts in the absence of antibiotic selection – thus hosts without plasmids were able to grow. The experiment ran for 600 generations and a bottleneck was used every 10 generations, diluting the population 1,000-fold. Due to time constraints and efficiency of the code, the simulation to mimic this experiment included a bottleneck every 7 generations, diluting the population approximately 100-fold. In P21, the cost of the plasmid was 40.4%, the estimated mutation rate was  $6.45 \times 10^{-4}$ , and the estimated loss rate was  $1.43 \times 10^{-5}$ . In H2, the cost of the plasmid was 14.9%, the mutation rate was  $6.45 \times 10^{-4}$ , and the loss rate was  $1.91 \times 10^{-9}$ . In R28, the plasmid cost was 5.4%, the mutation rate was  $6.45 \times 10^{-4}$ , and the loss rate was  $3.9 \times 10^{-2}$ . All of these values were taken directly from De Gelder *et al.* (2007).

The results from the simulation (Figure 14b) were compared to the results from the publication (Figure 14a). These two charts are very similar for plasmid pB10 in hosts P21 and R28, showing rapid plasmid loss. However, the simulation did not accurately portray the dynamic of the plasmid in host H2. It is believed that this was due to (i) the reported loss rate of the plasmid being orders of magnitude smaller than the reported loss rate for the plasmid in the other two hosts and (ii) the differences in population size between the laboratory experiment and the simulation. The reported loss rate of the plasmid pB10 in H2 was very small and the cost was not high enough to make up for the low loss rate and result in a rapid plasmid loss in the population. However, the trend that started to appear towards the end of the 100 generations in the simulation begins to match the trend seen in the laboratory experiment. Still, the similarity of the data from the simulation to two of the three laboratory experiments indicates that the simulation was able to mimic two of the three experimentally determined plasmid stability patterns the fairly successfully.



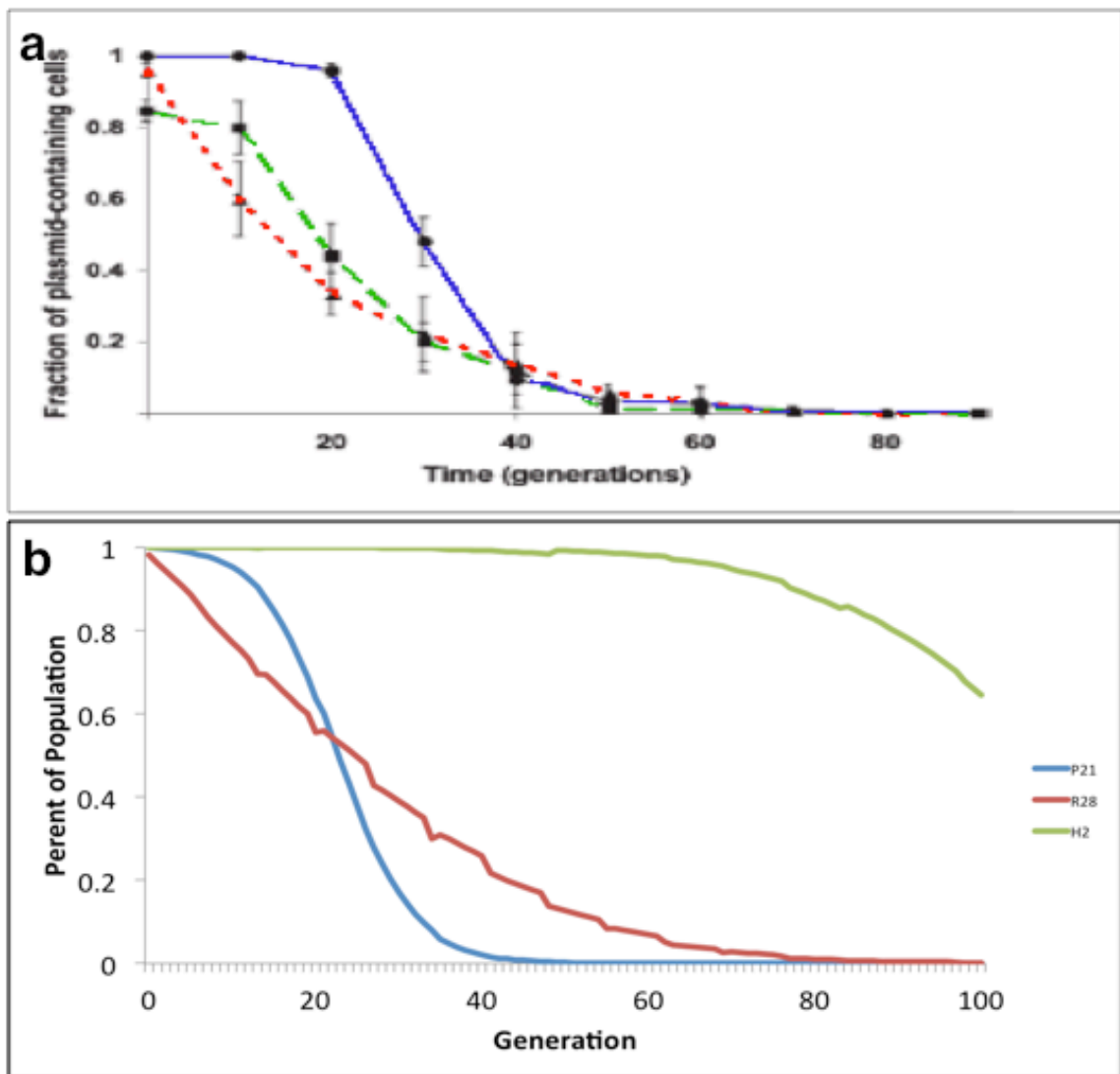


Figure 14: Simulation emulating De Gelder *et al.* (2007) study of plasmid pB10 in hosts P21, R28, and H2. (a) Graphical results from De Gelder *et al.*, 2007. This figure was artificially colored to match the results from the simulation. (b) Results from simulation.

The second chosen publication was from Hughes *et al.* (2012), using the 13.1 kb non-transferrable minireplicon pMS0506 in the host bacterium *Shewanella oneidensis* MR-1. This study focused on the evolutionary dynamics of pMS0506 in MR-1, observing the role of clonal interference in the population. The experiment ran for 400 generations and a bottleneck was used every 10 generations, diluting the population 1,000-fold. Again, due to time constraints and efficiency of the code, the simulation to mimic this experiment included a bottleneck every 7 generations, diluting the population

approximately 100-fold. This simulation was run for 400 generations. The cost of the plasmid was assumed to be 5%, the loss rate of the plasmid was  $2.5 \times 10^{-2}$ , and the mutation rate was estimated to be  $1.3 \times 10^{-3}$ . As this minireplicon is non-transferrable, the conjugation rate of the plasmid was set to 0.

The results from the simulation (Figure 15b) were then compared to the results from the laboratory experiment, reported in the publication from Hughes *et al.* (2012) (Figure 15a). In the simulation, one mutation in the replication region of the plasmid swept through the population and can be seen as dominating. Other mutations also occurred during the simulation, a total of 54, though due to limitations in the graphs, really only the top seven can be seen. The majority of these mutations occurred in the replication region of the plasmid, which is consistent with the mutations observed in the laboratory by Hughes *et al.* (2012).

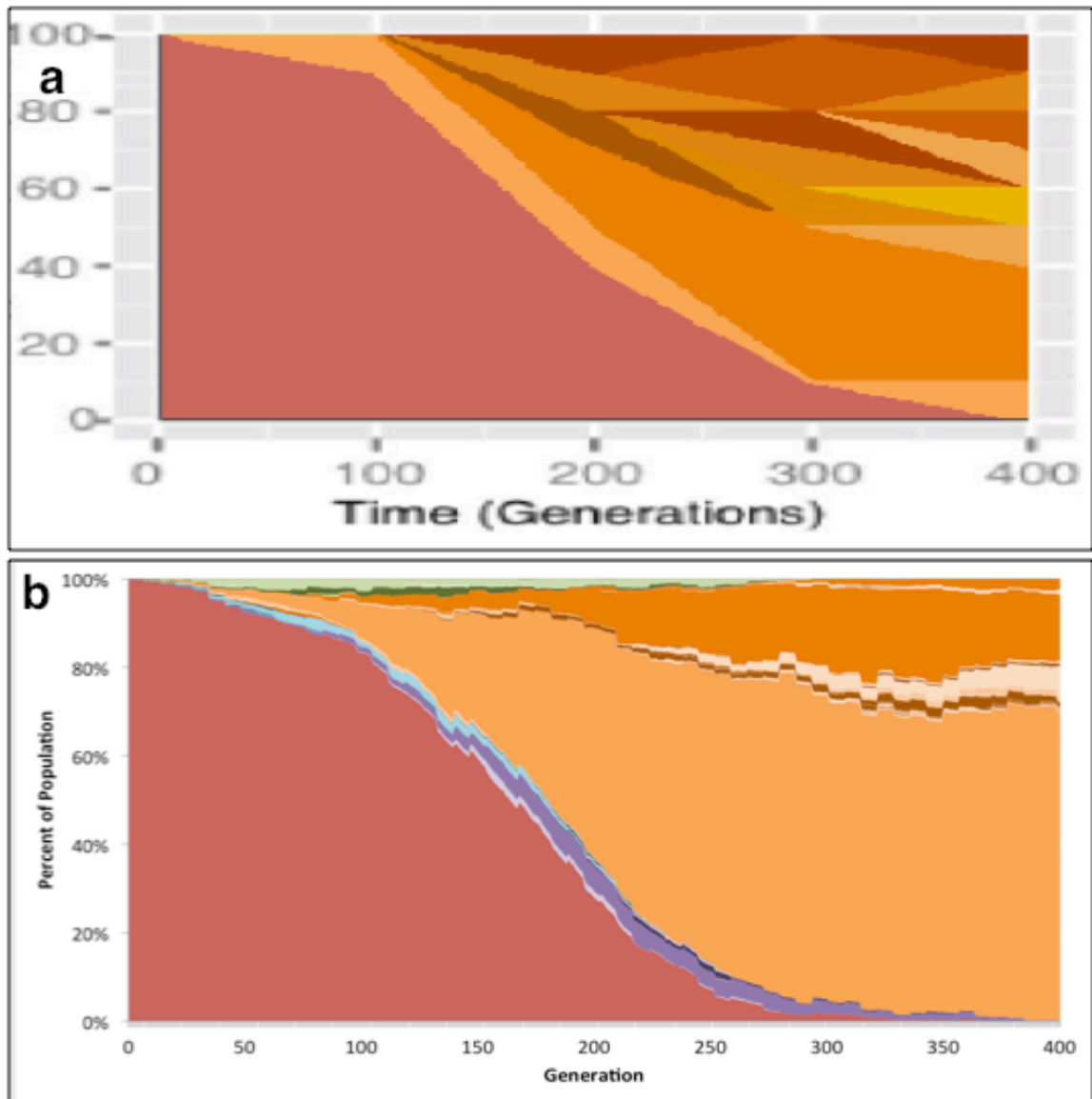


Figure 15: Simulation emulating dynamics of experimental evolution of minireplicon pMS0506 in host MR-1 in the presence of selection for the plasmid (Hughes *et al.*, 2012). (a) Results from Hughes *et al.* (2012). This figure was artificially colored to match the results from the simulation. (b) results from the simulation.

## Chapter 4: Discussion, Applications, and Future Work

### 4.1 Discussion

The model depicts a probabilistic simulation of plasmid evolution in bacteria. As is shown, changes in multiple parameters will affect the outcome in different manners. It can be seen that subtle or drastic changes in plasmid cost, mutation rate, conjugation rate, and loss rate can allow different mutants to sweep through the population and the varying times it takes for this to happen (Results sections 3.3-3.6).

With this simulation, it is possible to look at multiple factors over a large time range in an expedient manner. This is helpful in interpreting experimental results and generating hypotheses about underlying molecular mechanisms, which can be tested empirically. This model is useful in helping researchers determine experimental designs in pursuit of many different evolutionary questions – such as the likelihood of and time in which a plasmid could transfer to and stably reside in a novel host; the effect a fluctuating selection system could have on an antibiotic resistance plasmid; the length of time it will likely take a mutation to sweep through the population (depending on the population size); and which plasmid will likely “win” and sweep through the population if two (or more) are competing for the same resources.

The output of this model can be compared to published results (De Gelder *et al.*, 2007; Hughes *et al.*, 2012) showing that the statistical models developed within are sound and generate results very similar to those found in laboratory environments. This is incredibly important as it shows the model is statistically sound and can be used to help solidify hypotheses and direct research efforts.

This research has numerous important implications in antibiotic resistance research. Plasmids can cause many bacterial species to become multidrug resistant, thus limiting available treatment options. This simulation will enable the development of strategies to limit rapid horizontal spread of multidrug resistance by providing information on the

genetic determinants of plasmid evolution. One possible informational scenario could be the cost versus benefit analysis of a fluctuating antibiotic schedule, in which antibiotic treatment would stop for a certain number of days, while the cost of the plasmid limits host replication and results in more plasmid-free segregants. If the treatment is stopped for a number of days, the cost of the plasmid has the potential to outweigh its benefits and the hosts with plasmids may be out-competed by the plasmid-free segregants. When antibiotic treatment is then resumed, it will be easier to kill the host. This analysis will assist in determining the optimal number of days for the duration of antibiotic treatment. Other possible concentration- or time-dependent killing strategies (Sharma *et al.*, 2002) could also be simulated

## **4.2 Applications**

This model will be useful in creating hypotheses about plasmid biology. It will be useful in formulating ideas about plasmid host range changes, clonal interference, and providing information about antibiotic resistance in plasmids.

The computer model simulates plasmid mutation, loss, growth, and conjugation in a bacterial population. Information about the relative frequency of mutations in certain locations, the effects of these mutations, and correlations between different parameters can be gathered from the simulation to assist researchers in understanding the underlying evolutionary dynamics of plasmid diversity in a population.

Additionally, the model could be used to play with ranges of antibiotic concentrations. In this way, researchers could use the model to determine the minimum antibiotic concentration needed to kill a resistant bacterium. Furthermore, it could be used to look at the minimum selective concentration needed to enrich a resistant strain. It has been shown that antibiotic concentrations much lower than the inhibitory concentration can enrich resistant bacteria (Andersson and Hughes, 2012). Looking at a range of antibiotic concentrations, and the effects at differing points in that range, could prove useful in many different scenarios.

Viruses are another type of extra-chromosomal mobile genetic element. Viruses could be modeled in much the same way and this simulation could prove a starting point and/or reference for those models.

### **4.3 Future Work**

An important future goal of this project will be to add in a graphical user interface (GUI). Currently, the model is only in command line format. It would be easier for more researchers to be able to utilize this model if it included a GUI. Additionally, the GUI would be useful in explaining the input parameters and showing the default values of the simulation.

Currently, the model simulates a batch culture experiment performed in a lab. There is no spatial factor and hosts are assumed to always be in contact with one another. The model could be expanded, however, to look at a natural environment and formulate hypotheses about real-world plasmid dynamics. A spatial factor could be implemented, taking into account the possible distances between bacteria and how this may affect transfer of plasmids within a population through horizontal gene transfer. Plasmid conjugation requires physical contact between host cells. In a sparse population, where cells are only occasionally right next to each other, conjugation would occur much less frequently, due to the lag time between when contact with another cell could be established. A parameter to indicate the spread of cells would be useful in forming hypotheses about bacterial evolution in nature, as opposed to laboratory experiments. Other work in this area has been done by Krone *et al.*, (2007). Their team performs research in mathematically modeling plasmid conjugation and loss, where a spatial factor is taken into consideration. In their paper in *Microbiology*, Krone *et al.* (2007) devise an individual-based lattice model of plasmid dynamics is presented, which successfully simulates spatially structured habitats.

The model could be improved to be able to handle a larger population size. As it stands currently, it is not feasible to run the simulation with a large population size. An initial population size of 1,000 hosts is roughly the maximum of the simulation – when the

simulation grows to 100,000 hosts, the runs take an extremely long time to complete. At this stage, it took an average of over seven hours for one simulation to run with this population size. Larger populations will take longer – using a population size of a million took longer than 24 hours to complete one generation. When attempting to run the simulation for many generations, it is simply not economical to use this many hosts.

Furthermore, it would be a relatively simple task to implement two different hosts in this simulation. The model currently observes only one host and plasmid combination. However, if a second host were added, information about the evolution of a plasmid's host range could be gathered. Additionally, if a second plasmid was added and some intracellular competition taken into consideration, it would be possible to look at the competition of plasmids and the effect of antibiotic resistance on host fitness either under or free of selective pressure. Information on how mutations affect a plasmid's host range is incredibly useful in understanding the dynamics of plasmid spread in a population.

## References

- Andersson, D. I., and D. Hughes. "Evolution of antibiotic resistance at non-lethal drug concentrations". *Drug Resistance Updates* 15.3 (2012): 162-172.
- Ansari, Athar and Marc C. Gartenberg. "The Yeast Silent Information Regulator Sir4p Anchors and Partitions Plasmids". *Molecular and Cellular Biology* 17.12 (1997): 7061-7068.
- Bergstrom, Carl T., Marc Lipsitch, and Bruce R. Levin. "Natural Selection, Infectious Transfer and the Existence Conditions for Bacterial Plasmids". *Genetics* 155.2 (2000): 1505-1519.
- Betlach, M., V. Hershfield, L. Chow, H. Goodman, and HW Boyer. "A restriction endonuclease analysis of the bacterial plasmid controlling the *ecoRI* restriction and modification of DNA". *Federation Proceedings* 35.9 (1976): 2037-2043.
- Boe, L. and K. V. Rasmussen. "Suggestions as to quantitative measurements of plasmid loss". *Plasmid* 36.3 (1996): 153-159.
- Brown, Celese J., Diya Sen, Hirokazu Yano, Matthew L. Bauer, Linda M. Rogers, Geraldine A. Van der Auwera, and Eva M. Top. "Diverse Broad-Host-Range Plasmids from Freshwater Carry Few Accessory Genes". *Applied and Environmental Microbiology* 79.24 (2013): 7684-7695.
- Cangelosi, Angelo and Domenico Parisi. "Computer Simulation: A New Scientific Approach to the Study of Language Evolution". *Simulating the Evolution of Language* (2001).
- Canton, R. "Antibiotic resistance genes from the environment: a perspective through newly identified antibiotic resistance mechanisms in the clinical setting". *European Society of Clinical Microbiology and Infectious Diseases* 15.1 (2009): 20-25.
- Carattoli, Alessandra. "Resistance Plasmid Families in *Enterobacteriaceae*". *Antimicrobial Agents and Chemotherapy* 53.6 (2009): 2227-2238.
- Coe, E. H., Jr. "Some observations bearing on plasmid versus gene hypotheses for a conversion-type phenomenon". *Genetics* 46.7 (1961): 719-725.
- Dahlberg, Cecilia and Lin Chao. "Amelioration of the Cost of Conjugative Plasmid Carriage in *Escherichia coli* K12". *Genetics* 165.4 (2003): 1641-1649.
- De Gelder, Leen, José M. Ponciano, Zaid Abdo, Paul Joyce, Larry J. Forney, and Eva M. Top. "Combining Mathematical Models and Statistical Methods to Understand and Predict the Dynamics of Antibiotic-Sensitive Mutants in a Population of Resistant Bacteria During Experimental Evolution". *Genetics* 168.3 (2004): 1131-1144.
- De Gelder, Leen, José M. Ponciano, Paul Joyce, and Eva M. Top. "Stability of a promiscuous plasmid in different hosts: no guarantee for a long-term relationship". *Microbiology* 153.2 (2007): 452-463.
- De Gelder, Leen, Julia J. Williams, José M. Ponciano, Masahiro Sota, and Eva M. Top. "Adaptive Plasmid Evolution Results in Host-Range Expansion of a Broad-Host-Range Plasmid". *Genetics* 178.4 (2008): 2179-2190.



- del Solar, Gloria, Juan C. Alonso, Manuel Espinosa, and Ramón Díaz-Oreas. "Broad-host-range plasmid replication: an open question." *Molecular Microbiology* 21.4 (1996): 661-666.
- del Solar, Gloria, Rafael Giraldo, María Jesús Ruiz-Echevarría, Manuel Espinosa, and Ramón Díaz-Oreas. "Replication and Control of Circular Bacterial Plasmids". *Microbiology and Molecular Biology Reviews* 62.2 (1998): 434-464.
- del Solar, Gloria, and Manuel Espinosa. "Plasmid copy number control: an ever-growing story". *Molecular Microbiology* 37.3 (2000): 492-500.
- de Visser, J. Arjan G. M. and Daniel E. Rozen. "Clonal Interference and the Periodic Selection of New Beneficial Mutations in *Escherichia coli*". *Genetics* 172.4 (2006): 2093-2100.
- Dionisio, F., I. C. Conceição, A. C. R. Marques, L. Fernandes, and I. Gordo. "The evolution of a conjugative plasmid and its ability to increase bacterial fitness". *Biology Letters* 1.2 (2005): 250-252.
- Dougan, Gordon, J. H. Crosa, and Stanley Falkow. "Mobilization of the *Escherichia coli* Plasmid ColE1 (Colicin E1) and ColE1 Vectors Used in Recombinant DNA Experiments". *The Journal of Infectious Diseases* 137.5 (1978): 676-680.
- Fernández-López, Raúl, M. Pilar Garcillán-Barcia, Carlos Revilla, Miguel Lázaro, Luis Vielva, and Fernando de la Cruz. "Dynamics of the IncW genetic backbone imply general trends in conjugative plasmid evolution". *FEMS Microbiological Reviews* 30.6 (2006): 942-956.
- Foster, T. J. "Plasmid-Determined Resistance to Antimicrobial Drugs and Toxic Metal Ions in Bacteria". *Microbiological Reviews* 47.3 (1983): 361-409.
- Frost, Laura S., Raphael Leplae, Anne O. Summers, and Ariane Toussaint. "Mobile Genetic Elements: The Agents of Open Source Evolution". *Nature Reviews Microbiology* 3.9 (2005): 722-732.
- Hawkey, P. M., P. M. Bennett, and Carol A. Hawkey. "Cryptic Plasmids in Hospital Isolates of *Providencia stuarti*". *Journal of Medical Microbiology* 18.2 (1984): 277-284.
- Hernández, Carmen Sara, Clara Martínez, Manuel Porcar, Primitivo Caballero, and Juan Ferré. "Correlation between serovars of *Bacillus thuringiensis* and type I  $\beta$ -exotoxin production". *Journal of Invertebrate Pathology* 82.1 (2003): 57-62.
- Hughes, Julie M. "Evolutionary dynamics of plasmid-host adaptation". Master's thesis. University of Idaho, Moscow, Idaho. (2011).
- Hughes, Julie M., Brian K. Lohman, Gail E. Deckert, Eric P. Nichols, Matt Settles, Zaid Abdo, and Eva M. Top. "The Role of Clonal Interference in the Evolutionary Dynamics of Plasmid-Host Adaptation". *American Society for Microbiology* 3.4 (2012).
- Huovinen, Pentti. "Trimethoprim Resistance". *Antimicrobial Agents and Chemotherapy* 31.10 (1987): 1451-1456.
- Jensen, Rasmus Bugge, and Kenn Gerdes. "Mechanism of DNA segregation in prokaryotes – ParM partitioning protein of plasmid R1 co-localizes with its

- replicon during the cell cycle". *European Molecular Biology Organization Journal* 18.4 (1999): 4076-4084.
- Kado, Clarence I. "Origin and evolution of plasmids". *Antonie van Leeuwenhoek* 73.1 (1998): 117-126.
- Karr, Jonathan R., Jayodita C. Sanghvi, Derek N. Macklin, Miriam V. Gutschow, Jared M. Jacobs, Benjamin Bolival Jr., Nacyra Assad-Garcia, John I. Glass, and Markus W. Covert. "A Whole-Cell Computational Model Predicts Phenotype from Genotype". *Cell* 150.2 (2012): 389-401.
- Khan, Saleem A. "Rolling-Circle Replication of Bacterial Plasmids". *Microbiology and Molecular Biology Reviews* 61.4 (1997): 442-455.
- Klaenhammer, Todd R. and Rosemary B. Sanozky. "Conjugal Transfer from *Streptococcus lactis* ME2 of Plasmids Encoding Phage Resistance, Nisin Resistance and Lactose-fermenting Ability: Evidence for a High-frequency Conjugative Plasmid Responsible for Abortive Infection of Virulent Bacteriophage". *Journal of General Microbiology* 131.6 (1985): 1531-1541.
- Krone, Stephen M., Ruinan Lu, Randal Fox, Haruo Suzuki, and Eva M. Top. "Modelling the spatial dynamics of plasmid transfer and persistence". *Microbiology* 153.8 (2007): 2803-2816.
- Kunnimalaiyaan, Muthusamy, David M. Stevenson, Yansheng Zhou, and Patricia S. Vary. "Analysis of the replicon region and identification of an rRNA operon on pBM400 of *Bacillus megaterium* QM B1551". *Molecular Microbiology* 39.4 (2001): 1010-1021.
- Lawson, Michael A. "The Antibiotic Resistance Problem Revisited". *The American Biology Teacher* 70.7 (2008): 405-410.
- Lederberg, Joshua. "Cell Genetic and Hereditary Symbiosis". *Physiological Review* 32.4 (1952): 403-430.
- Lederberg, Joshua. "Plasmid (1952-1997)". *Plasmid* 39.1 (1998): 1-9.
- Lenski, Richard E., Suzanne C. Simpson, and Toai T. Nguyen. "Genetic Analysis of a Plasmid-Encoded, Host Genotype-Specific Enhancement of Bacterial Fitness". *Journal of Bacteriology* 176.11 (1994): 3140-3147.
- Levin, Bruce R., Véronique Perrot, and Nina Walker. "Compensatory Mutations, Antibiotic Resistance and the Population Genetics of Adaptive Evolution in Bacteria". *Genetics* 154.3 (2000): 985-997.
- Levy, Stuart B. "Antibiotic Resistance: Consequences of Inaction". *Clinical Infectious Diseases* 33.Supplement\_3 (2001): S124-S129.
- Liu, Hong, Jun-Jie Zhang, Su-Jun Wang, Xian-En Zhang, and Ning-Yi Zhou. "Plasmid-borne catabolism of methyl parathion and *p*-nitrophenol in *Pseudomonas* sp. strain WBC-3". *Biochemical and Biophysical Research Communications* 334.4 (2005): 1107-1114.
- Lobo, Ingrid. "Pleiotropy: One Gene Can Affect Multiple Traits". *Nature Education* 1 (2008): 10.

- Lowe, David E., and Ian J. Glomski. "Cellular and Physiological Effects of Anthrax Exotoxin and Its Relevance to Disease". *Frontiers in Cellular and Infection Microbiology* 2.76 (2012): 1-13.
- Magnuson, Roy David. "Hypothetical Functions of Toxin-Antitoxin Systems". *Journal of Bacteriology* 189.17 (2007): 6089-6092.
- Martinez, Betsy, Jeffrey Tomkins, Lawrence P. Wackett, Rod Wing, and Michael J. Sadowsky. "Complete Nucleotide Sequence and Organization of the Atrazine Catabolic Plasmid pADP-1 from *Pseudomonas* sp. Strain ADP". *Journal of Bacteriology* 183.19 (2001): 5684-5697.
- Martínez-Lavanchy, Paula M., Christina Müller, Ivonne Nijenhuis, Uwe Kappelmeyer, Marieke Buffing, Kyle McPherson, and Hermann J. Heipieper. "High Stability and Fast Recovery of Expression of the TOL Plasmid-Carried Toluene Catabolism Genes of *Pseudomonas putidamt-2* under Conditions of Oxygen Limitation and Oscillation". *Applied and Environmental Microbiology* 76.20 (2010): 6715-6723.
- Menn, Fu-Min, Bruce M. Applegate, and Gary S. Sayler. "NAH Plasmid-Mediated Catabolism of Anthracene and Phenanthrene to Naphthoic Acids". *Applied and Environmental Microbiology* 59.6 (1993): 1938-1942.
- Modi, Rajiv I., and Julian Adams. "Coevolution in Bacterial-Plasmid Populations". *Evolution* 45.3 (1991): 656-667.
- Neu, Harold C. "The Crisis in Antibiotic Resistance". *Science* 257.5073 (1992): 1064-1073.
- Norman, Anders, Lars H. Hansen, and Søren J. Sørensen. "Conjugative plasmids: vessels of the communal gene pool". *Philosophical Transactions of the Royal Society* 364.15 (2009): 2275-2289.
- Novick, Richard P., and Mark H. Richmond. "Nature and Interactions of the Genetic Elements Governing Penicillinase Synthesis in *Staphylococcus aureus*". *Journal of Bacteriology* 90.2 (1965): 467-480.
- Novick, Richard P. "Plasmid Incompatibility". *Microbiological Reviews* 51.4 (1987): 381-395.
- Palumbi, Stephen R. "Humans as the World's Greatest Evolutionary Force". *Science* 293.5536 (2001): 1786-1790.
- Plasmid* 1.1 (1977).
- Prusinkiewicz, Przemyslaw. "Modeling plant growth and development". *Current Opinion in Plant Biology* 7.1 (2004): 79-83.
- R Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>.
- Richmond, Mark H. "Dominance of the Inducible State in Strains of *Staphylococcus aureus* Containing Two Distinct Penicillinase Plasmids". *Journal of Bacteriology* 90.2 (1965): 370-374.
- Rimšeliénė, R., R. Vaišvila, and A. Janulaitis. "The *eco72IC* gene specifies a *trans*-acting factor which influences expression of both DNA methyltransferase and endonuclease from the *Eco72I* restriction-modification system". *Gene* 157.1-2 (1995): 217-219.

- Roth, Thomas F., and Donald R. Helinski. "Evidence for Circular DNA Forms of a Bacterial Plasmid". *Proceedings of the National Academy of Sciences of the United States of America* 58.2 (1967): 650-657.
- RStudio (2012). RStudio: Integrated development environment for R (Version 0.96.122) [Computer software]. Boston, MA. Available from <http://www.rstudio.org/>.
- Schmidt, Torsten, Karl Friehs, and Erwin Flaschel. "Rapid determination of plasmid copy number". *Journal of Biotechnology* 49.1-3 (1996): 219-229.
- Scott, June R. "Regulation of Plasmid Replication". *Microbiological Reviews* 48.1 (1984): 1-23.
- Sen, Diya, Geraldine A. Van der Auwera, Linda M. Rogers, Christopher M. Thomas, Celeste J. Brown, and Eva M. Top. "Broad-Host-Range Plasmids from Agricultural Soils Have IncP-1 Backbones with Diverse Accessory Genes". *Applied and Environmental Microbiology* 77.22 (2011): 7975-7983.
- Sengupta, Manjitha, and Stuart Austin. "Prevalence and Significance of Plasmid Maintenance Functions in the Virulence Plasmids of Pathogenic Bacteria". *Infection and Immunity* 79.7 (2011): 2502-2509.
- Sharma, K. K., H. Sangraula, P. K. Mediratta. "Some new concepts in antibacterial drug therapy". *Indian Journal of Pharmacology* 34.6 (2002): 390-396.
- Slater, Frances R., Mark J. Bailey, Adrian J. Tett, and Sarah L. Turner. "Progress towards understanding the fate of plasmids in bacterial communities". *FEMS Microbiological Ecology* 66.1 (2008): 3-13.
- Smillie, Chris, M. Pilar Garcillán-Barcia, M. Victoria Francia, Eduardo P. C. Rocha, and Fernando de la Cruz. "Mobility of Plasmids". *Microbiology and Molecular Biology Reviews* 74.3 (2010): 434-452.
- Sompolinsky, David, Miriam Ben-Yakov, Mordehai Aboud, and Ida Boldur. "Transferable resistance factors with mutator effect in *Salmonella typhi*". *Mutation Research – Fundamental and Molecular Mechanisms of Mutagenesis* 4.2 (1967): 119-127.
- Sota, Masahiro, Hirokazu Yano, Julie M. Hughes, Gary W. Daughdrill, Zaid Abdo, Larry J. Forney, and Eva M. Top. "Shifts in the host range of a promiscuous plasmid through parallel evolution of its replication initiation protein". *The International Society for Microbial Ecology Journal* 4.12 (2010): 1568-1580.
- Spangler, Rudolph, Shiping Zhang, Judy Krueger, and Geoffrey Zubay. "Colicin Synthesis and Cell Death". *Journal of Bacteriology* 163.1 (1985): 167-173.
- Spellberg, Brad, Robert Guidos, David Gilbert, John Bradley, Helen W. Boucher, W. Michael Scheld, John G. Bartlett, and John Edwards Jr. "The Epidemic of Antibiotic-Resistant Infections: A Call to Action for the Medical Community from the Infectious Diseases Society of America". *Clinical Infectious Diseases* 46.2 (2008): 155-164.
- Strahilevitz, Jacob, George A. Jacoby, David C. Hooper, and Ari Robicsek. "Plasmid-Mediated Quinolone Resistance: a Multifaceted Threat". *Clinical Microbiology Reviews* 22.4 (2009): 664-689.
- Suarez, Juan E., and Maria C. Mendoza. "Plasmid-encoded fosfomycin resistance". *Antimicrobial Agents and Chemotherapy* 35.5 (1991): 791-795.

- Summers, David. "Timing, self-control and a sense of direction are the secrets of multicopy plasmid stability". *Molecular Microbiology* 29.5 (1998): 1137-1145.
- Thavasi, R, K. Aparnadevi, S. Jayalakshmi, and T. Balasubramanian. "Plasmid mediated antibiotic resistance in marine bacteria". *Journal of Environmental Biology* 28.3 (2007): 617-621.
- Tomita, Haruyoshi, Koichi Tanimoto, Satoshi Hayakawa, Kyoko Morinaga, Kohji Ezaki, Hisaji Oshima, and Yasuyohsi Ike. "Highly Conjugative pMG1-Like Plasmids Carrying Tn1546-Like Transposons That Encode Vancomycin Resistance in *Enterococcus faecium*". *Journal of Bacteriology* 185.23 (2003): 7024-7028.
- Van Melderen, Laurence and Manuel Saavedra De Bast. "Bacterial Toxin-Antitoxin Systems: More Than Selfish Entities?" *PLOS Genetics* 5.3 (2009): e1000437.
- Wooley, John C. and Herbert S. Lin. "Computational Modeling and Simulation as Enablers for Biological Discovery". *Catalyzing Inquiry at the Interface of Computing and Biology* (2005).
- World Health Organization. *Antimicrobial Resistance: Global report on surveillance*. 2014.
- Zhong, Xue, Jason Droesch, Randal Fox, Eva M. Top, and Stephen M. Krone. "On the meaning and estimation of plasmid transfer rates for surface-associated and well-mixed bacterial populations". *Journal of Theoretical Biology* 294 (2012): 144-152.
- Zielenkiewicz, Urszula, and Piotr Ceglowski. "Mechanisms of plasmid stable maintenance with special focus on plasmid addiction systems". *Acta Biochimica Polonica* 48.4 (2001): 1003-1023.

## Appendix

### Code

```

### REGULAR SIMULATION
rm(list=ls())

### Set the working directory
MAIN_DIR = '~/Documents/Simulation/Profile'
setwd(MAIN_DIR)

### Create classes of mutation, plasmid, and host
setClass("mutation", representation(type="character",
location="character", size="numeric", effect="numeric"))

setClass("plasmid", representation(name="character",
attribute="character", my_mutation="list", inc="character",
loss_rate="numeric",
conj_rate="numeric", mut_rate="numeric", cost="numeric"))

setClass("host", representation(number="character",
type="character", alive="logical", growth_rate="numeric",
num_grows="numeric",
have_plasmid="logical", plasmid="list", parent="character",
generation="numeric",
death="character", conjugation_partner="character",
conjugation_gen="character", received_plasmid="numeric",
plasmid_time="list",
name="character"))

setClass("counts", representation(type="character",
number="numeric"))

### Parameters
host_type <-c("A") # Host name(s)
plasmid_type <- c("R") # Plasmid name(s)
mut_loc_prob <- c(.25,.25,.25,.25) # Probability for where
a mutation is likely to occur

lin = 1 # Number of times to repeat entire simulation
init_pop = 1000 # Initial population size
rep = 10 # Generations, iterations of the code
bottleneck_gen = 7 # Bottleneck after this many iterations
bottleneck_pop = init_pop # Bottleneck to this population
size

```

```

pop_limit = 100 # How big the population can get (initial
population size * population limit)

mutation_type <- c("Insertion","Deletion","Pt Mutation") #
Types of possible mutations
LocName <-
c("Replication","Stability","Conjugation","Accessory") #
Mutation locations
mutate <- c("mutate","Don't mutate")
DEBUG = 0

### Output files lists
total_results = list()
number_mutations = list()
type_mutations = list()
number_plasmids = list()
hosts_alive = list()
name_hosts = list()

for(lineages in 1:lin){

  hosts <- list()
  growth = 0
  host_cnt = 0
  num_recip = 0
  prev_num_recip = 0
  num_don = 0
  conjugation = 0
  inc_counter = 0

  init_mutation = list()
  init_plasmid = list()

  pop = init_pop
  density = pop

  #Initialize hosts
  for(i in 1:pop){
    host = host_type
    # Growth rate in gen/day
    growth_rate = 1

    init_mutation = list()
    init_plasmid = list()
    host_name = paste("host_",i,sep="")

```

```

        init_host <- new("host", name=host_name, type=host,
number=as.character(host_cnt), alive=TRUE,
growth_rate=growth_rate, num_grows = 0, have_plasmid=FALSE,
plasmid=init_plasmid, parent="NA", generation=0,
death="NA", conjugation_partner="NA", conjugation_gen="NA")
        host_cnt = host_cnt + 1
        hosts[[host_name]] = init_host
    }

#Simulate plasmid distribution
for(i in 1:pop){
    #Assign plasmids to hosts
    PlasmidType = plasmid_type

    attribute = "Antibiotic Resistant"
    hosts[[i]]@have_plasmid = TRUE
    inc = "P"

    cost = 0.05
    mut_rate = 0.0005
    conj_rate = 0.0001
    loss_rate = 0.005

    assign_plasmid <- new("plasmid", name=PlasmidType,
attribute=attribute, my_mutation=init_mutation, inc=inc,
loss_rate=loss_rate, conj_rate=conj_rate,
mut_rate=mut_rate, cost=cost)
    hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)+1]]
= assign_plasmid
}

# we only conjugate into hosts without a plasmid, so,
the prob of conjugation is equal to rate * num_recip
# technically, this should always be 0 at the beginning
for(p in 1:length(hosts)){
    if(hosts[[p]]@have_plasmid == FALSE){
        num_recip = num_recip + 1
    }
}

for(j in 0:rep){
    # Bottleneck
    if (j %% bottleneck_gen == 0){
        if (length(hosts) >= init_pop){
            new_hosts = sample(hosts, init_pop)

```



```

        hosts = list()
        hosts = new_hosts
        if(DEBUG >= 1){
            cat('Bottleneck: length(hosts) =',
length(hosts))
        }
        } else {
            hosts = hosts
        }
    }

    pop = length(hosts)
    density = pop
    prev_num_recip = num_recip
    num_recip = 0

    cat('Length hosts (', length(hosts), ') Rep (', j,
') Lin (', lineages,')\n')
    hosts = sample(hosts,length(hosts))
    if(DEBUG >= 2){
        cat('Just sampled hosts. Length of hosts is:',
length(hosts), '\n')
    }

    for(q in 1:length(hosts)){
        i = hosts[[q]]@name
        grow_cnt = 0
        if(DEBUG >= 2){
            cat('Working on host:', i, '\n')
        }

        # if it has a plasmid the host can grow and/or
conjugate
        if(length(hosts[[i]]@plasmid) >= 1){
            num_don = num_don + 1

            # for each plasmid we can grow and/or
conjugate
            for(k in 1:length(hosts[[i]]@plasmid)){
                if((hosts[[i]]@growth_rate)*(1-
hosts[[i]]@plasmid[[k]]@cost) > 1){
                    grow_action == 0
                } else if((hosts[[i]]@growth_rate)*(1-
hosts[[i]]@plasmid[[k]]@cost) < 0){

```

```

        grow_action == 1
    } else{
        grow_action =
sample(0:1,1,prob=c((hosts[[i]]@growth_rate)*(1-
hosts[[i]]@plasmid[[k]]@cost),1-
(hosts[[i]]@growth_rate)*(1-hosts[[i]]@plasmid[[k]]@cost)))
    }

    # we check this a lot because we had
problems during dev
    if(length(hosts[[i]]@plasmid) >= 1){
        if(grow_action == 0){

if(hosts[[i]]@plasmid[[k]]@attribute == "Antibiotic
Resistant"){

                                #GROW if less than pop limit
and it has a plasmid that is Antibiotic Resistant
                                if(length(hosts) <
pop_limit*init_pop){
                                new_host_name =
paste('host_',host_cnt,sep='')
                                hosts[[new_host_name]]
= hosts[[i]]
hosts[[new_host_name]]@name = new_host_name
hosts[[new_host_name]]@parent = hosts[[i]]@number
                                host_cnt = host_cnt + 1

hosts[[new_host_name]]@plasmid[[k]]@my_mutation =
hosts[[i]]@plasmid[[k]]@my_mutation
                                if (DEBUG >= 2){
                                    cat('Growing\n')
                                }

hosts[[new_host_name]]@generation = j

                                # keep track of how
many grows I've done
                                hosts[[i]]@num_grows =
hosts[[i]]@num_grows + 1

                                # If host grows,
possibility of plasmid loss

```

```

if(hosts[[i]]@plasmid[[k]]@loss_rate > 1){
    loss = 1
} else
if(hosts[[i]]@plasmid[[k]]@loss_rate < 0){
    loss = 0
} else{
    loss =
sample(0:1,1,prob=c(1-
hosts[[i]]@plasmid[[k]]@loss_rate,hosts[[i]]@plasmid[[k]]@l
oss_rate))
}

if (loss == 1){
    if(DEBUG>=1){
        cat('Host: ',
pop+1, ' lost plasmid.\n')
    }

hosts[[new_host_name]]@plasmid = init_plasmid

if(length(hosts[[new_host_name]]@plasmid) == 0){

hosts[[new_host_name]]@have_plasmid = FALSE
    if (DEBUG >=
2){
        cat('(loss
== 1) test of hosts[[' ,new_host_name, ']]:\n')

cat('\thave_plasmid=', hosts[[new_host_name]]@have_plasmid)

cat('\tlength(plasmid)=' ,
length(hosts[[new_host_name]]@plasmid),'\n')
    }
} else {
    cat('ERROR:
Loss did not remove the plasmid.')
    stop()
}
}

if (DEBUG >= 1){
    cat('\tIncremented
population (', length(hosts), ') Rep (', j, ')\n ')
}
pop = pop + 1

```

```

density = density + 1
grow_cnt = grow_cnt + 1
} else {
  if (DEBUG >= 1){
    cat('length(hosts)
>= 1000. Not growing.\n')
  }
}

if(length(hosts[[i]]@plasmid) >= 1){
  #mutate?
  mutation =
sample(0:1,1,prob=c(1-
hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@mut_rate,h
osts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@mut_rate))
  if (mutation == 1){
    if(DEBUG >= 1){

cat('Mutating\n')
    }
    type =
sample(mutation_type,1,prob=c(0.30,0.30,0.40))

    #Using rate and
type, calculate mutation size and location
    if (type ==
"Insertion"){
      if(DEBUG >= 2){

cat('Insertion\n')
      }
      size =
sample(1:50,1)
    } else if (type ==
"Deletion"){
      if(DEBUG >= 2){

cat('Deletion\n')
      }
      size =
sample(1:50,1)
    } else{
      if(DEBUG >= 2){
        cat('Point
Mutation\n')
      }
    }
  }
}

```

```

}
size = 0
}

location =
effect =
if(location ==
    if(DEBUG >= 2){
}
# Mutation in
# Can also have
if (effect >
1){

hosts[[i]]@plasmid[[k]]@loss_rate =
hosts[[i]]@plasmid[[k]]@loss_rate / effect

hosts[[i]]@plasmid[[k]]@cost = hosts[[i]]@plasmid[[k]]@cost
/ effect

incompatibility = sample(0:1,1,prob=c(0.95,0.05))

if(incompatibility == 1){

hosts[[i]]@plasmid[[k]]@inc = "NA"

inc_counter = inc_counter + 1
}
} else if

(effect < 1){

if(hosts[[i]]@plasmid[[k]]@loss_rate == 0){

hosts[[i]]@plasmid[[k]]@loss_rate = (-1 * effect) / 100
} else{

hosts[[i]]@plasmid[[k]]@loss_rate =
hosts[[i]]@plasmid[[k]]@loss_rate * (-1 * effect)
}
}

```

```

if(hosts[[i]]@plasmid[[k]]@cost == 0){
hosts[[i]]@plasmid[[k]]@cost = (-1 * effect) / 100
} else{

hosts[[i]]@plasmid[[k]]@cost = hosts[[i]]@plasmid[[k]]@cost
* (-1 * effect)
}
} else{
cat('effect
== 0 in replication region.\n')
}
} else if (location
== "Conjugation"){
if(DEBUG >= 2){
cat('Conjugation\n')
}
# Mutation in
this region may affect plasmid's conjugation rate
if (effect >
1){
hosts[[i]]@plasmid[[k]]@conj_rate =
hosts[[i]]@plasmid[[k]]@conj_rate * effect
} else if
(effect < 1){
hosts[[i]]@plasmid[[k]]@conj_rate =
hosts[[i]]@plasmid[[k]]@conj_rate * (-1 * effect)
} else {
cat('effect
== 0 in conjugation location.\n')
}
} else if (location
== "Stability"){
if(DEBUG >= 2){
cat('Stability\n')
}
# Mutation in
this region may affect plasmid's loss rate
if (effect >
1){

```

```

hosts[[i]]@plasmid[[k]]@loss_rate =
hosts[[i]]@plasmid[[k]]@loss_rate / effect

incompatibility = sample(0:1,1,prob=c(0.95,0.05))

if(incompatibility == 1){

hosts[[i]]@plasmid[[k]]@inc = "NA"

inc_counter = inc_counter + 1
}
} else if
(effect < 1){

if(hosts[[i]]@plasmid[[k]]@loss_rate == 0){

hosts[[i]]@plasmid[[k]]@loss_rate = (-1 * effect) / 100
} else{

hosts[[i]]@plasmid[[k]]@loss_rate =
hosts[[i]]@plasmid[[k]]@loss_rate * (-1 * effect)
}
} else {
cat('effect
== 0 in stability region.\n')
}
} else {
if(DEBUG >= 2){

cat('Accessory\n')

}
# Accessory

region
# Mutation in
this region may affect plasmid's accessory genes
# Can also have
an effect on the overall cost of the plasmid
if (effect >
1){

hosts[[i]]@plasmid[[k]]@cost = hosts[[i]]@plasmid[[k]]@cost
/ effect
} else if
(effect < 1){

```

```

                                                                    if(effect <
-8){
if(hosts[[i]]@plasmid[[k]]@cost == 0){
hosts[[i]]@plasmid[[k]]@cost = (-1 * effect) / 100
                                                                    } else{
hosts[[i]]@plasmid[[k]]@cost = hosts[[i]]@plasmid[[k]]@cost
* (-1 * effect)
                                                                    }
                                                                    if
(hosts[[i]]@plasmid[[k]]@attribute == "Antibiotic
Resistant"){
hosts[[i]]@plasmid[[k]]@attribute = "NA"
                                                                    }
                                                                    }else{
if(hosts[[i]]@plasmid[[k]]@cost == 0){
hosts[[i]]@plasmid[[k]]@cost = (-1 * effect) / 100
                                                                    } else{
hosts[[i]]@plasmid[[k]]@cost = hosts[[i]]@plasmid[[k]]@cost
* (-1 * effect)
                                                                    }
                                                                    }
                                                                    } else {
                                                                    cat('effect
== 0 in accessory region.\n')
                                                                    }
                                                                    }
                                                                    }
                                                                    # Create new
mutation using calculated values
                                                                    new_mutation <-
new("mutation", type=type, location=location, size=size,
effect=effect)
                                                                    curr_mutation <-
hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@my_mutatio
n
hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@my_mutatio
n[[length(hosts[[i]]@plasmid[[k]]@my_mutation)+1]] <-
new_mutation

```



```

mutation"
} else {
  # No mutation
  type = "No

  size = 0
  location = "None"
}
}
} else{
  if(DEBUG >= 2){
    cat('Not growing, no
antibiotic resistance')
  }
} else{
  if (DEBUG >= 2){
    cat('Not growing')
  }
}
}

if(length(hosts[[i]]@plasmid) >= 1){
if(hosts[[i]]@plasmid[[k]]@conj_rate*prev_num_recip >= 1){
  conj_action = 1
} else{
  conj_action =
sample(0:1,1,prob=c(1-
hosts[[i]]@plasmid[[k]]@conj_rate*prev_num_recip,hosts[[i]]
@plasmid[[k]]@conj_rate*prev_num_recip))
}

if(conj_action == 1){
  conjugation = conjugation + 1
  conj_cnt = 0
  #conjugate
  if(DEBUG >= 2){
    cat("Conjugating\n")
  }

  #conj_cnt = 0
  found_partner = FALSE
  repeat{
    partner <-
sample(hosts,1,replace=FALSE)

```

```

p_index, "\n")
p_index = -1
p_index = partner[[1]]@name
#cat("Got partner name: ",
which(unlist(lapply(hosts, function(input)
partner[[1]]@number %in% input@number)))
#cat(p_index, " ",
partner[[1]]@number, '\n')

#cat("hosts[p_index]@conjugation_partner",
hosts[[p_index]]@conjugation_partner, '\n')

#cat('partner@conjugatin_partner',
partner@conjugation_partner, '\n')
if(p_index == -1){
  cat('ERROR: unable to
find index of partner')
  break
} else{
  if( DEBUG >= 2){
    cat('partner_index
= ', p_index, '\n')
  }
}
conj_cnt = conj_cnt + 1
# Try a reasonable number
of times to find a partner
if(conj_cnt <= pop){
  #if able to conjugate
  (acceptable partner), break out of loop
}
if(length(partner[[1]]@plasmid) == 0){
  found_partner =
TRUE
  break
}
# No suitable partner
found-
} else {
  found_partner = FALSE
  break
}
}
if(found_partner){

```

```

                                prev_num_recip =
prev_num_recip - 1

hosts[[p_index]]@conjugation_partner = hosts[[i]]@number
hosts[[p_index]]@conjugation_gen = as.character(j)
hosts[[p_index]]@received_plasmid = j

                                if (DEBUG >= 2){
                                    cat('(pre) Number of
plasmids:', length(partner[[1]]@plasmid), '\n')
                                }

                                # NOTE: Must perform deep
copy here- for some reason not all of the class members
were being copied.

                                new_plasmid =
new("plasmid",
name=hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@name,

attribute=hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@
attribute,

my_mutation=hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]
@my_mutation,

inc=hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@inc,

loss_rate=hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@
loss_rate,

conj_rate=hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@
conj_rate,

mut_rate=hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@m
ut_rate,

cost=hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@cost)

                                if (DEBUG >= 2){
                                    cat('Found partner\n')
                                    cat('Created new
plasmid:')
                                }

```

```

#cat('Adding plasmid to "',
hosts[[p_index]]@type, '"', Num: ', hosts[[p_index]]@number,
' in rep:', j, '\n')

hosts[[p_index]]@plasmid[[length(hosts[[p_index]]@plasmid)+
1]] = new_plasmid

hosts[[p_index]]@plasmid_time[[length(hosts[[p_index]]@plasmid_time)+1]] = rep

if(length(hosts[[p_index]]@plasmid) >= 1){

hosts[[p_index]]@have_plasmid = TRUE
}

if (DEBUG >= 2){
cat('(post) Number of
plasmids:', length(partner[[1]]@plasmid), '\n')
cat('(found_partner)
test of hosts[['i,']]:\n')
cat('\thave_plasmid=',
partner[[1]]@have_plasmid)
cat('\tlength(plasmid)=', length(partner[[1]]@plasmid))
} else {
cat('Couldnt find partner
in...',conj_cnt,'tries\n')
prev_num_recip = 0
#stop()
}

#mutate?
mutation =
sample(0:1,1,prob=c(1-
hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@mut_rate,
hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@mut_rate))
if (mutation == 1){
if(DEBUG >= 1){
cat('Mutating\n')
}
type =
sample(mutation_type,1,prob=c(0.30,0.30,0.40))

```

```

#Using rate and type,
calculate mutation size and location
    if (type == "Insertion"){
        if(DEBUG >= 2){
            cat('Insertion\n')
        }
        size = sample(1:50,1)
    } else if (type ==
"Deletion"){
        if(DEBUG >= 2){
            cat('Deletion\n')
        }
        size = sample(1:50,1)
    } else{
        if(DEBUG >= 2){
            cat('Point
Mutation\n')
        }
        size = 0
    }

    location =
sample(LocName,1,prob=mut_loc_prob)
    effect = runif(1,min=-
10,max=10)
    if(location ==
"Replication"){
        if(DEBUG >= 2){
            cat('Replication\n')
        }
        # Mutation in this
region may affect plasmid's loss rate
        # Can also have an
effect on the overall cost of the plasmid
        if (effect > 1){
            hosts[[i]]@plasmid[[k]]@loss_rate =
            hosts[[i]]@plasmid[[k]]@loss_rate / effect

            hosts[[i]]@plasmid[[k]]@cost = hosts[[i]]@plasmid[[k]]@cost
            / effect

            incompatibility =
sample(0:1,1,prob=c(0.95,0.05))
            if(incompatibility
== 1){

```

```

hosts[[i]]@plasmid[[k]]@inc = "NA"
inc_counter =
inc_counter + 1
}
} else if (effect < 1){

if(hosts[[i]]@plasmid[[k]]@loss_rate == 0){

hosts[[i]]@plasmid[[k]]@loss_rate = (-1 * effect) / 100
} else{

hosts[[i]]@plasmid[[k]]@loss_rate =
hosts[[i]]@plasmid[[k]]@loss_rate * (-1 * effect)
}

if(hosts[[i]]@plasmid[[k]]@cost == 0){

hosts[[i]]@plasmid[[k]]@cost = (-1 * effect) / 100
} else{

hosts[[i]]@plasmid[[k]]@cost = hosts[[i]]@plasmid[[k]]@cost
* (-1 * effect)
}
} else{
cat('effect == 0 in
replication region.\n')
}
} else if (location ==
"Conjugation"){
if(DEBUG >= 2){

cat('Conjugation\n')
}
# Mutation in this
region may affect plasmid's conjugation rate
if (effect > 1){

hosts[[i]]@plasmid[[k]]@conj_rate =
hosts[[i]]@plasmid[[k]]@conj_rate * effect
} else if (effect < 1){

hosts[[i]]@plasmid[[k]]@conj_rate =
hosts[[i]]@plasmid[[k]]@conj_rate * (-1 * effect)
} else {

```

```

                                cat('effect == 0 in
conjugation location.\n')
                                }

                                } else if (location ==
"Stability"){
                                if(DEBUG >= 2){
                                    cat('Stability\n')
                                }
                                # Mutation in this
region may affect plasmid's loss rate
                                if (effect > 1){

hosts[[i]]@plasmid[[k]]@loss_rate =
hosts[[i]]@plasmid[[k]]@loss_rate / effect
                                incompatibility =
sample(0:1,1,prob=c(0.95,0.05))
                                if(incompatibility
== 1){

hosts[[i]]@plasmid[[k]]@inc = "NA"
                                inc_counter =
inc_counter + 1
                                }
                                } else if (effect < 1){

if(hosts[[i]]@plasmid[[k]]@loss_rate == 0){

hosts[[i]]@plasmid[[k]]@loss_rate = (-1 * effect) / 100
                                } else{

hosts[[i]]@plasmid[[k]]@loss_rate =
hosts[[i]]@plasmid[[k]]@loss_rate * (-1 * effect)
                                }
                                } else {
                                    cat('effect == 0 in
stability region.\n')
                                }
                                } else {
                                    if(DEBUG >= 2){
                                        cat('Accessory\n')
                                    }
                                    # Accessory region
                                    # Mutation in this
region may affect plasmid's accessory genes

```

```

# Can also have an
effect on the overall cost of the plasmid
    if (effect > 1){
hosts[[i]]@plasmid[[k]]@cost = hosts[[i]]@plasmid[[k]]@cost
/ effect
    } else if (effect < 1){
    if(effect < -8){

if(hosts[[i]]@plasmid[[k]]@cost == 0){

hosts[[i]]@plasmid[[k]]@cost = (-1 * effect) / 100
    } else{

hosts[[i]]@plasmid[[k]]@cost = hosts[[i]]@plasmid[[k]]@cost
* (-1 * effect)
    }
    if
(hosts[[i]]@plasmid[[k]]@attribute == "Antibiotic
Resistant"){

hosts[[i]]@plasmid[[k]]@attribute = "NA"
    }
    }else{

if(hosts[[i]]@plasmid[[k]]@cost == 0){

hosts[[i]]@plasmid[[k]]@cost = (-1 * effect) / 100
    } else{

hosts[[i]]@plasmid[[k]]@cost = hosts[[i]]@plasmid[[k]]@cost
* (-1 * effect)
    }
    }
    } else {
cat('effect == 0 in
accessory region.\n')
    }
}

# Create new mutation using
calculated values
new_mutation <-
new("mutation", type=type, location=location, size=size,
effect=effect)

```



```

curr_mutation <-
hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@my_mutation
n

hosts[[i]]@plasmid[[length(hosts[[i]]@plasmid)]]@my_mutation
n[[length(hosts[[i]]@plasmid[[k]]@my_mutation)+1]] <-
new_mutation

    }else {
        if(DEBUG >= 2){
            cat('Not mutating in
conjugation\n')
        }
    }
} else{
    if(DEBUG >= 2){
        cat('Not conjugating\n')
    }
}
}
} else{
    num_recip = num_recip + 1
}

pop = length(hosts)
density = pop

if(DEBUG >= 2){
    cat("Grow count = ", grow_cnt, "\n")
}
}

cat('Length hosts (', length(hosts), ') Rep (', j,
') Lin (', lineages,')\n')

# OUTPUT TO FILE

# Print

cat('Parameters\n')
cat('\tInitial Population = ', init_pop, '\n')
cat('\tGenerations = ', rep, '\n')
cat('\tLineages = ', lin, '\n')
cat('\tLoss Rate = ', loss_rate, 'Conjugation Rate
= ', conj_rate, 'Mutation Rate = ', mut_rate, 'Cost = ',
cost, '\n')

```

```

        cat('\tBottleneck after ', bottleneck_gen,
'generations and divide population by', bottleneck_pop,
'\n')
        cat('\tPopulation limit is ', pop_limit, 'times the
population\n')
        cat('\tMutation Location Probabilities = ',
mut_loc_prob, 'for regions ', LocName, 'repectfully\n')
        cat('\tGrowth Rate = ', growth_rate, '\n')
        cat('\tIncompatibility mutations =', inc_counter,
'\n')
        for(x in 1:length(hosts)){
            cat('\nHost:', hosts[[x]]@number, ", " ,
hosts[[x]]@type)
            cat('\n')
            cat('\tParent:', hosts[[x]]@parent)
            cat('\n')
            cat('\tGeneration:', hosts[[x]]@generation)
            cat('\n')
            cat('\tViable:', hosts[[x]]@alive)
            cat(', Non-viability:', hosts[[x]]@death)
            cat('\n')
            cat('\tNum Divisions:', hosts[[x]]@num_grows,
"Replication Rate:", hosts[[x]]@growth_rate)
            cat('\n')
            cat('\tReceived Plasmid:',
hosts[[x]]@received_plasmid)
            cat('\n')
            cat('\tPlasmid (', length(hosts[[x]]@plasmid),
'): \n')
                if(length(hosts[[x]]@plasmid) >= 1){
                    for(y in 1:length(hosts[[x]]@plasmid)){
                        cat('\t\tType:',
hosts[[x]]@plasmid[[y]]@name)
                            cat(', Attribute:',
hosts[[x]]@plasmid[[y]]@attribute)
                                cat(', Incompatibility:',
hosts[[x]]@plasmid[[y]]@inc)
                                    cat(', Cost:',
hosts[[x]]@plasmid[[y]]@cost)
                                        cat('\n')
                                            cat('\t\tLoss Rate:',
hosts[[x]]@plasmid[[y]]@loss_rate)
                                                cat(', Mutation Rate:',
hosts[[x]]@plasmid[[y]]@mut_rate)
                                                    cat(', Conjugation Rate:',
hosts[[x]]@plasmid[[y]]@conj_rate)

```

```

        cat('\n')
        cat('\t\tMy Mutations (',
length(hosts[[x]]@plasmid[[y]]@my_mutation), '):\n')

if(length(hosts[[x]]@plasmid[[y]]@my_mutation) >= 1){
    for(z in
1:length(hosts[[x]]@plasmid[[y]]@my_mutation)){
        cat('\t\t\tType:',
hosts[[x]]@plasmid[[y]]@my_mutation[[z]]@type)
        cat(', Location:',
hosts[[x]]@plasmid[[y]]@my_mutation[[z]]@location)
        cat(', Effect:',
hosts[[x]]@plasmid[[y]]@my_mutation[[z]]@effect)
        cat('\n')
    }
}
        cat('\t\tMutationLocation (',
length(hosts[[x]]@plasmid[[y]]@my_mutation), '): ')

if(length(hosts[[x]]@plasmid[[y]]@my_mutation) >= 1){
    for(z in
1:length(hosts[[x]]@plasmid[[y]]@my_mutation)){
        cat(', ',
hosts[[x]]@plasmid[[y]]@my_mutation[[z]]@location)
    }
}
        cat('\t\tEndMutations')
    }
}
        cat('\tConjugation:',
hosts[[x]]@conjugation_partner)
        cat(', Generation:',
hosts[[x]]@conjugation_gen)
        cat('\n\n')
    }

    length(hosts)
    #results

    sink()
}
}
closeAllConnections()
)

```