Convergent mutation rates can be used to estimate the rate of homologous recombination in bacteria

W. Zane Billings[1*], Kara Schatz[2*], Jerry Yang[3*], Johnathan Rowell[4], Louis-Marie Bobay[5†]

[1]Department of Biology, Western Carolina University; [2]Department of Mathematics and Computer Science, Xavier University, [3]Department of Molecular and Cellular Biology, University of California, Berkeley, [4]Department of Mathematics and Statistics, University of North Carolina, Greensboro, [5]Department of Biology, University of North Carolina, Greensboro

Abstract

Bacteria are commonly regarded as clonal organisms, inheriting genetic information from a single parent cell. However, evidence shows that bacteria undergo homologous recombination, in which they incorporate foreign DNA into their genomes. Recombination is not yet well understood as several studies have come to inconsistent conclusions regarding the rate at which it occurs. But homoplasies, which are sections of shared DNA not inherited from a common ancestor, are easier to detect, and they arise from either recombination or convergent mutation. Thus, estimating convergent mutations allows us to indirectly infer the rate of recombination. Employing a probabilistic model verified by simulation, we can obtain measurements of the recombination rate for several different bacterial species. Ultimately, an accurate metric for recombination rate provides a better understanding of bacterial evolution and adaptation.

Keywords: homologous recombination, bacteria, microbial evolution, genomics, modeling

Introduction

Recombination in the context of bacteria refers to any process through which foreign genomic material is incorporated into the genome. This typically occurs by one of three mechanisms: transformation, transduction, or conjugation. During transformation, a bacterium uptakes genetic material from the environment, which is often left behind by dead bacteria which have been lysed [1–3]. In transduction, a viral vector injects its genetic material into the bacterium, and the genetic material is integrated into the bacterial genome [4]. Finally, conjugation is the process through which bacteria have the ability to "mate," where one bacteria forms a sex pilus between itself and another bacterium and sends duplicated genetic material across the structure [5]. The donor bacterium must be fertility factor, or F-factor, positive, possessing an episome which can be present as either a plasmid or integrated into the bacterial chromosome. The F-factor episome encodes proteins necessary for the construction of the sex pilus, and in the process of conjugation the acceptor bacterium will typically receive the F-factor [5].

After the acceptor bacterium receives the DNA, several mechanisms exist to incorporate the exogenous material into the genome, most notably the Rec-A pathway, where the foreign genetic material is used to repair breaks in DNA [6–9]. The new DNA strand essentially replaces a section of the old genome. In contrast, nonhomologous recombination, or end joining, occurs when DNA is inserted into the genome and the genome becomes longer [10]. We have elected to focus solely on homologous recombination events due to the added difficulty of comparing genomes with the possibility of changing length over multiple generations.

All bacterial species were once thought to be entirely clonal: based on serotypes of infectious bacteria, microbiologists noted that only certain serotypes caused disease, and the clonal concept of bacteria was extrapolated from the serotypic data, since identical serotypes were found in geographically isolated regions. [11, 12]. More recent research has demonstrated that many bacteria are in fact sexual on some level [12], existing on a spectrum from clonality, where no recombination occurs, to panmixis, a situation in which all members of a bacterial population have the ability to recombine and share genetic information. Panmictic

*These authors contributed equally to this work.
†Corresponding author–email: ljbobay@uncg.edu

species such as *Neisseria gonorrhoeae* are effectively sexual due to the high levels of recombination that occur. Recombination in clonal species is much more rare. Because recombination rate can vary significantly between species, no clear metric currently exists for consistently and effectively calculating the rate of recombination for different bacterial species [13].

The reason for bacterial recombination is not entirely understood. Muller's Ratchet, the Red Queen Hypothesis, and Clonal Inference are the three major hypotheses proposed to explain recombination. Muller's Ratchet claims that purely clonal bacteria would, due to genetic drift, accumulate deleterious mutations until these mutations become lethal, leading to a loss of genetic variation and consequential loss of fitness [14–16]. Recombination allows bacteria to replace deleterious segments, and thus potentially alleviate negative selective pressures. The Red Queen Hypothesis states that organisms must continually evolve to remain competitive with other organisms in order to prevent extinction [17]. Finally, the clonal interference hypothesis posits that some clonal organisms with distinct, beneficial mutations on different gene loci would compete with each other and have less fitness individually than if all genes were concentrated in one organism [18–20]. Together, these explain how recombination increases fitness by allowing for the coalescence of many mutated, beneficial alleles in one organism, allowing for a greater fitness of the population in a shorter amount of time. [21].

A consistent metric for recombination rates does not yet exist. While statistical tests exist and are held to be accurate, especially when used in conjunction, these tests can only detect homoplasic sites, and determine if recombination is occurring; they do not provide a rate [12, 22–25]. Attempts to measure recombination rates have been made [13, 26–28]

The consistent ability to detect the presence of recombination leads directly to the question of how rates of recombination can be determined. Methods for calculating a rate are much rarer than the methods proposed simply to detect recombination. Analysis of empirical MLST data provided a foundation to build up a method on [13, 26]. Additionally, methods from population genetics have been utilized to estimate recombination rates [27]. Since all of these data were collected and processed in different ways, comparison between rates calculated by different methods can be difficult to do and to interpret.

Thus, we propose an indirect measurement of recombination rates, utilizing detectable homoplasies and a probabilistic model to reach an estimate. A homoplasy is any section of DNA which is present in multiple extant strains of bacteria, but is not derived from common ancestry. Homoplasic sites can arise in two ways: by convergent mutation, in which two genomes coincidentally mutate to the same nucleotide in the same position, or by recombination. Since homoplasies are created solely convergent mutations and recombination events, the number of homoplasic sites is equal to the sum of the number of convergent mutation sites and the number of recombinant sites. The total number of homoplasic sites between DNA strands is relatively easy to calculate. If we model the expected number of convergent mutations between two strands, we can calculate the number of homoplasies, and then subtract the number of expected convergent mutations from the number of homoplasies to obtain the number of recombinant sites. In other words: $h = r + c$, where h is the number of homoplasic sites, r the number of recombinant sites, and c the number of convergent sites.

Once all three numbers are obtained, the ratio of recombination events to mutation events, r/m is trivial to find and provides an "effective recombination rate." An effective recombination rate, ρ, gives information about the relative prevalence of recombination compared to mutation in a species of bacteria, and ignores "silent recombination" events, where a recombinant sequence is exactly identical to the previous sequence which was replaced. So, ρ contains relative information about how often recombination occurs and can be used as an estimator for the amount of recombination occurring within a species.

Our proposed model estimates the number of convergent mutations between two strands of DNA. The model takes into account the likelihood of different mutations (e.g. transition and transversion probabilities), assumes mutations are Poisson distributed, and accounts for multiple mutation events at a single site. Using our model, we can estimate the number of recombination events that have occurred within any given bacterial species, which in conjunction with the number of homoplasies allows us to estimate the relative impact of recombination compared to mutation on divergence in the population, giving us an effective recombination rate for a given species.

Methods

add a generic description of the methods section here

Variables

Here we list all of the variables that will be used.

c	number of convergent mutation sites
c_q	number of convergent mutations shared between q strains
g	number of generations over which the DNA has been replicated
h	number of total homoplasies
h_c	number of homoplasies due to convergent mutation
L	number of base pairs in the DNA sequence
l_i	length of branch i on the phylogenetic tree
m_i	number of mutations in strain i
m_i^*	number of mutation sites in strain i
n	number of strains in the bacterial population
r	number of recombinant sites
x_i	expected number of mutations at a single site in strain i
α	probability of a transition
β	probability of a transversion
μ	mutation rate in mutations per base pair per generation
μ^*	mutation rate in mutations per base pair
σ	number of overlapping mutation sites
ϕ	probability of a transversion to the complementary base
κ	ratio of transitions to transversions; $\frac{\alpha}{\beta\phi+\beta(1-\phi)} = \frac{\alpha}{\beta}$

Simple Model

To begin, we determined the probability of a convergent mutation assuming that no factors aside from strict probability would have any impact. This resulted in equation **??**, which has four parts:

1. the probability of m_1 mutations in strain 1
2. the probability of m_2 mutations in strain 2
3. the probability of σ overlapping mutation sites given m_1 and m_2
4. the probability of c convergent mutations sites given σ

These values come from equations **??**, **??**, **??**, and **??**, respectively. Note that we define overlapping mutation sites as those sites along the DNA strand at which a mutation has occurred in both strain 1 and strain 2, but the nucleotides present there may be the same or different. Convergent mutation sites are those overlapping sites for which the nucleotides are the same. The product of these four probabilities gives the probability of seeing c convergent mutations given m_1 and m_2. Because we do not actually know the values of m_1 and m_2, we find the sum over all possible values of m_1, m_2, and σ, which gives the total probability of c convergent mutations in any situation. We let m_1, m_2, and σ range anywhere from 0 to L. Even though seeing L mutations or L overlapping mutation sites will likely never happen from a probabilistic standpoint, we include it so that our model truly covers all possibilities.

This model relies on the following assumptions:

1. The number of mutations that occur on a strand is distributed according to the Poisson distribution
   (a) Mutations are independent events
   (b) Mutations are rare
   (c) The mutation rate is constant
   (d) Mutations do not occur simultaneously
2. The probability of mutation to any nucleotide is equivalent

3. Each nucleotide site can be mutated at most once
4. The mutation rate is constant over all generations
5. The probability of a mutation does not change over the course of multiple generations

Probability of m_i Mutations

First, the probability of m_1 and m_2 are given by equation **??**, which is simply the probability according to a Poisson distribution. The Poisson distribution is applicable here because mutations meet all the assumptions of the Poisson distribution: they are independent, rare events that occur with a constant rate and cannot occur simultaneously. Mutations are definitely independent and rare events. While they can occur simultaneously, the probability they happen in immediate succession is much higher. We assume that the time in between two mutations may be arbitrarily small, which allows the assumption to be satisfied. The constant rate at which mutations occur is μ. Thus, each of the requirements are met. The Poisson distribution is well-established and accurate for mutation events (CITE THIS), so we use the Poisson probability mass function to estimate the probability of m_i:

$$P(m_i) = e^{-\lambda}\frac{\lambda^{m_i}}{m_i!},$$

where λ is the average or expected number of events in a time interval. In the context of mutations, $\lambda = \mu^* L$ (effective mutation rate times strand length) since $\mu^* L$ is the expected number of mutations along a strand in one generation.

Probability of σ Overlapping Mutation Sites

The probability of σ overlapping sites is given by equation **??**, which is a minor modification of the combinatorial concept known as the multinomial coefficient. The multinomial coefficient can be interpreted as the number of ways to put distinct items into a certain number of distinct categories with a given number of items desired in each category. In our case, the multinomial coefficient is:

$$\binom{L}{\sigma, m_1 - \sigma, m_2 - \sigma, L - m_1 - m_2 + \sigma},$$

which is the number of ways to place L items (each of the nucleotide sites) into four different categories: overlapping mutation sites, mutation sites only on strain 1, mutation sites only on strain 2, and non-mutating sites given that we want σ items, $m_1 - \sigma$ items, $m_2 - \sigma$ items, and $L - m_1 - m_2 + \sigma$ items in each category, respectively. These values arise because σ is the number of overlapping mutation sties, $m_1 - \sigma$ is the number of remaining mutation sites in strain 1 that are not overlapping with strain 2, $m_2 - \sigma$ is the number of remaining mutation sites in strain 2 that are not overlapping with strain 1, and $L - (\sigma - (m_1 - \sigma) - (m_2 - \sigma)) = L - m_1 - m_2 + \sigma$ is the remaining number of sites that do not mutate. Thus, these are the number of sites from L that fit each of the categories.

This multinomial coefficient represents the number of ways that we can arrange m_1 and m_2 mutations such that we get σ overlapping sites. These are the "successful" ways. However, there are actually

$$\binom{L}{m_1}\binom{L}{m_2}$$

total ways to arrange the mutations along the strands. Thus, if we want to find the probability of seeing σ overlapping sites, then we must find the proportion of the total ways that are "successful," which is given by:

$$P(\sigma \mid m_1, m_2, L) = \frac{\binom{L}{\sigma, m_1 - \sigma, m_2 - \sigma, L - m_1 - m_2 + \sigma}}{\binom{L}{m_1}\binom{L}{m_2}}$$

This equation gives us the probability of arranging m_1 and m_2 and getting σ, which is equivalent to the probability of getting σ overlapping sites given m_1 and m_2.

Probability of c Convergent Mutation Sites

Finally, the probability of c convergent mutation sites is given by equation **??**. Since convergent sites are a subset of overlapping sites, we assume that we know the number of overlapping sites and simply find the number of those such that the nucleotides in strain 1 and 2 would match. As the number of convergent mutations can be described in terms of success and failure, where a convergent overlapping site is a success and a non-convergent overlapping site is a failure, the probability of a given number of convergent mutations can be determined by the binomial probability density function, which requires three parameters: n (the number of trials), k (the desired number of successes), and p (the probability of a success). In our case, $n = \sigma$ and $k = c$ because we want to have c convergent mutation sites ("successes") out of a total of σ overlapping sites ("trials"). Our probability of success is the probability of two DNA strands mutating at the same site to the same nucleotide.

$$\Pi(c \mid \sigma) = \binom{\sigma}{c}(p)^c(1-p)^{\sigma-c}$$

To simplify the model, we assume that the strands are identical to start. Then, both strands must mutate to the same of the remaining three nucleotides. Again to simplify the model, we assume that each nucleotide is just as likely to be the result of a mutation. Therefore, there is a $1/3$ chance of mutating to each of the other nucleotides. The probability of both strands mutating to the same nucleotide is $1/3 * 1/3 = 1/9$, since these are independent events. However, there are three different nucleotides that could arise and lead to a convergent mutation. Therefore, the total probability of a convergent mutation over all the three scenarios is: $1/9 + 1/9 + 1/9 = 1/3$. Thus, the probability of c convergent mutation sites given σ overlapping mutation sites is given by the binomial probability density function with $n = \sigma$, $k = c$, and $p = 1/3$, which is:

$$\Pi(c \mid \sigma) = \binom{\sigma}{c}\left(\frac{1}{3}\right)^c\left(1-\frac{1}{3}\right)^{\sigma-c}$$

This equation required the key assumption that the probability of mutation to each nucleotide is equivalent. Unfortunately, this is not a realistic assumption to make. Thus, we later revamped this equation to make the necessary accommodations that make it applicable to a more general case with parameters that can be modified.

Complete Simple Model

Now, putting each of those four parts together comprises the original simple version of our model. The probability of c convergent mutations is simply the product of the four probabilities that have just been explained. There is one last step, and that is figuring out the total probability. Of course there are many different scenarios that can generate c convergent mutations. For example, let's say we want to know the probability of 2 convergent mutations occurring. Well, it is possible that this could occur with $m_1 = 2$, $m_2 = 2$, and $\sigma = 2$. It is also possible that this could occur with $m_1 = 50$, $m_2 = 50$, and $\sigma = 10$. In order to find the total probability, we must find the sum of the probabilities of each scenario that could produce these 2 convergent mutations. The same holds for any number of convergent mutations. To find this, we use a triple sum over all possible values of σ, m_1, and m_2, which is from 0 to L for each. This allows us to arrive at our complete model:

$$P(c) = \sum_{\sigma=0}^{L}\sum_{m_1=0}^{L}\sum_{m_2=0}^{L}\text{Poisson}(m_1)\text{Poisson}(m_2)P(\sigma \mid m_1, m_2, L)\Pi(c \mid \sigma)$$

This is simply the probability of convergent mutations, but what we really want is the number that we expect to occur. To get this, we use the well known method of calculating expected value by taking the sum of all possible values multiplied by their respective probabilities:

$$E[c] = \sum_{c=0}^{L} cP(c)$$

Simulation Verification

To validate our model, we programmed a simulation in Python that follows the Monte Carlo method. The simulation takes random DNA strands and mutates them according to the following five inputs from the user:

1. L = length of the DNA strands (in base pairs)
2. μ = mutation rate (in mutations per base pair per generation)

Upon termination, the simulation outputs the total number of convergent mutations that have occurred between two DNA strands that originated from a common ancestor.

The simulation begins by randomly generating a single 'ancestor' strand of DNA of length L and duplicating this strand so that there are 2 identical daughter strands. Then, it accesses each nucleotide in the strand and randomly generates a "new" nucleotide to replace the old one. The "new" nucleotide is not necessarily completely new as there is always a chance that no mutation will occur and the nucleotide will remain the same. The nucleotide choices are made based on a random weighted choice, where the probability of no mutation is $1 - \mu$, and the probability of a mutation to eahc of the other nucleotides is $1/3$. For example, if the nucleotide starts as A, then the simulation generates a "new" nucleotide where the probabilities of generating A, T, G, and C are $1 - \mu, 1/3, 1/3, and 1/3$, respectively. This "new" nucleotide replaces the original. Once this process has occurred for each nucleotide on each strand, the program goes through each nucleotide site tallying up the number of sites for which strand 1 and strand 2 have the same nucleotide that does not match that of the ancestor strand. These are the convergent mutations. Finally, the program has counted the total number of convergent mutations that have arisen between any two DNA strands, and it outputs this value.

To verify our model, we ran this simulation for 1000 iterations for various values of L and μ. Following the Monte Carlo method, we took the running average of the number of convergent mutations as the iterations ran and compared this to the expected number of convergent mutations predicted by our model.

Accommodating Varying Rates of Particular Nucleotide Substitutions

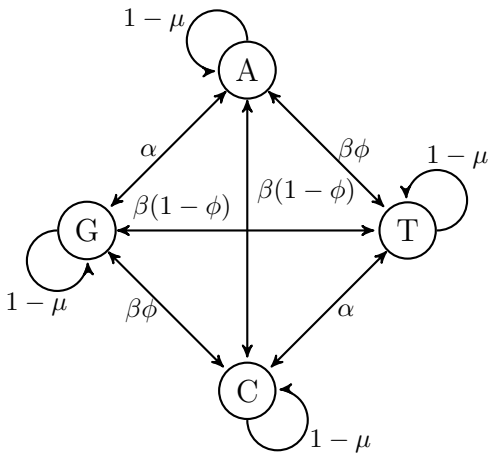
As mentioned earlier, the actual likelihood of mutating to each of the other three nucleotides is not equivalent. Instead a discrepancy arises between transitions and transversions. Transitions are mutations where the original nucleotide changes within its same class (purine to purine or pyrimidine to pyrimidine), whereas transversions are mutations in which the nucleotide changes to the opposite class (purine to pyrimidine or vice versa). Evidence has shown that transitions tend to be more common than transversions due to the similar structures encountered with transitions [29].

Fortunately, incorporating this characteristic only requires us to modify equation **??**. This was the binomial probability density function with $n = \sigma$, $k = c$, and $p = 1/3$. Now, $n = \sigma$ and $k = c$ is still accurate, but it is definitely inaccurate to use $p = 1/3$ as the probability of a convergent mutation. So, we need to determine a new value for p.

To do so, we use κ to represent the ratio of transitions to transversions. If we set κ equal to α/β, then we can define α as the probability of a transition and β as the probability of a transversion. Then, $\alpha + \beta$ is the total probability of a mutation, which is equivalent to our previous μ value. Values of κ and μ have been estimated for mutations, so we can do some algebraic transformations to get:

$$\alpha = \frac{\mu\kappa}{\kappa + 1} \text{ and } \beta = \frac{\mu}{\kappa + 1}$$

Now, for each nucleotide, there is one possible transition and two possible transversions, so we also need a way to distinguish between the different transversions. We define ϕ as the probability of a transversion to its complementary base; then, $1 - \phi$ becomes the probability of transversion to the other base. The following graph captures these mutations and their probabilities for every possible case:



The same information can be presented in matrix form as follows:

$$\begin{matrix} A & T & G & C \end{matrix}$$
$$\begin{bmatrix} 1-\mu & \beta\phi & \alpha & \beta(1-\phi) \\ \beta\phi & 1-\mu & \beta(1-\phi) & \alpha \\ \alpha & \beta(1-\phi) & 1-\mu & \beta\phi \\ \beta(1-\phi) & \alpha & \beta\phi & 1-\mu \end{bmatrix} \begin{matrix} A \\ T \\ G \\ C, \end{matrix}$$

where column i represents the starting nucleotide and row j represents the one to which it mutates. In other words, the (i, j) entry is the probability of a mutation from i to j.

With these probabilities, we can modify our original $\Pi(c \mid \sigma)$ function so that the value of p (the probability of success) in the binomial probability mass distribution reflects these new values, but the rest stays the same.

Instead of a $1/3$ chance of a mutation to each nucleotide in each strand, we have an α chance of a convergent mutation due to a transition in each strand, a $\beta\phi$ chance due to one of the two transversions, and a $\beta(1-\phi)$ chance due to the other transversion. In our Π function, we assume that there are σ sites that mutate in both strands. Thus, we are really interested in these probabilities out of the total probability that a mutation occurred, which is $\alpha + \beta\phi + \beta(1-\phi) = \alpha + \beta$. Since mutations are independent, the probability of the same type of mutation occurring in both strands is simply the probability in one strand squared. Thus, the total probability of a convergent mutation of any type is sum of these probabilities squared, which becomes our new p value:

$$p = \left(\frac{\alpha}{\alpha+\beta}\right)^2 + \left(\frac{\beta\phi}{\alpha+\beta}\right)^2 + \left(\frac{\beta(1-\phi)}{\alpha+\beta}\right)^2$$

Since κ is more widely used as a metric for the ratio of transitions to transversions, and it requires less variables, we rewrote p in terms of κ and ϕ alone:

$$p = \frac{\kappa^2 + 2\phi^2 - 2\phi + 1}{(\kappa+1)^2}$$

Using this value of p gives us a more general formula of our Π function, which we denote as $\overline{\Pi}$. This function is given by the following:

$$\overline{\Pi}(c \mid \sigma) = \binom{c}{\sigma} * \left(\frac{\kappa^2 + 2\phi^2 - 2\phi + 1}{(\kappa+1)^2}\right)^c * \left(1 - \frac{\kappa^2 + 2\phi^2 - 2\phi + 1}{(\kappa+1)^2}\right)^{\sigma - c}$$

Including this new version makes our overall model more representative of what occurs when bacteria mutate, so it is more accurate. For now, we use $\kappa = 3$ and $\phi = 1/2$, but these values can very easily be modified as we gain more information about the comparative likelihoods of mutation to each other nucleotide [29]

Again, our model does rely on some inherent assumptions; these are listed below:

1. The number of mutations the occur on a strand is distributed according to the Poisson distribution
   (a) Mutations are independent events
   (b) Mutations are rare
   (c) The mutation rate is constant
   (d) Mutations do not occur simultaneously
2. The probabilities of each transition are equivalent
3. The probabilities of each transversion to the complementary base pair are equivalent
4. Each nucleotide site can be mutated at most once
5. The mutation rate is constant over all generations
6. The probability of a mutation does not change over the course of multiple generations
7. Codon bias is not present
8. Selection pressure is not present

Simulation Verification

Again, to validate our model, we compared the results to those from our simulation. We first had to update our simulation to reflect the changes that were made to our model. To do so, the simulation requires two additional parameters:

1. κ = ratio of transitions to transversions
2. ϕ = probability of transversion to a nucleotide's complementary base pair

The simulation still outputs the total number of convergent mutations that have occurred between two strands of DNA that originated from the same ancestor, but now those mutations occur with different probabilities.

The simulation runs in the same manner as before. The only change is that now, the nucleotide choices are made based on a random weighted choice, where the weights are those described earlier: $1-\mu$, α, $\beta\phi$, and $\beta(1-\phi)$. For example, if the nucleotide starts as A, then the simulation generates a "new" nucleotide where the probabilities of generating A, T, G, and C are $1-\mu, \beta\phi, \alpha, and \beta(1-\phi)$, respectively.

Again, we ran the simulation for 1000 iterations for various values of L and μ, and compared the running average to the expected value outputted by our model.

Accommodating Generations

At this point, our model was still inaccurate due to the fact that it assumed the probability of a convergent mutation would not change over several generations. As more generations pass, the probability of a convergent mutation does in fact change. If we instead allow multiple generations to pass, then there are more "paths" a nucleotide site can take to arrive at its terminal nucleotide. For example, if a certain nucleotide site mutates from A to T over the course of 2 generations, then it did not necessarily mutate directly from A to T during one of the generations. Instead, it could have mutated from A to G in the first generation and then from G to T in the following generation. Likewise, it could have mutated from A to C in the first generation and then from C to T in the following. Since there are more possible ways to arrive at the terminal nucleotide, the probability of doing so changes. Again, the only change that must be made to our model is to the probability of a convergent mutation in equation **??**.

We use the mutation matrix mentioned previously to determine the probability of a convergent mutation over multiple generations. With this matrix, we know the probabilities of each nucleotide arising. Now, if we raise this matrix to the power of g representing the number of generations that have passed, then we will still have the probability of each different mutation (or lack thereof) occurring, but these probabilities span g generations. In other words, the (i, j) entry will still represent the probability of mutating from i to j, but now this does not have to happen in a single generation; it happens sometime over the course of g

generations. Then, we have the following mutation matrix:

$$\begin{bmatrix} 1-\mu & \beta\phi & \alpha & \beta(1-\phi) \\ \beta\phi & 1-\mu & \beta(1-\phi) & \alpha \\ \alpha & \beta(1-\phi) & 1-\mu & \beta\phi \\ \beta(1-\phi) & \alpha & \beta\phi & 1-\mu \end{bmatrix}^g$$

The probability of a convergent mutation can be found as follows. If we start at any given nucleotide, there are three others to which it can mutate. Recall that equation **??** assumes that a mutation has occurred, so we do not need to consider the case that a nucleotide stays the same. The probability of ending at one of the other nucleotides after g generations is given by one of the following matrix entries: $M^g_{ij_1}, M^g_{ij_2}, {M^g_{ij_3}}^2$, where i is the initial nucleotide, and j_1, j_2, and j_3 are the other three nucleotides. The total probability of a site having a different nucleotide in the last generation than in the first generation is $M^g_{ij_1} + M^g_{ij_2} + M^g_{ij_3}$; this is the total probability of an uncorrected mutation after g generations. (If a mutation occurs, then there is always a chance that a random point mutation will return it back to its original nucleotide; this is a corrected mutation). Now, we have the following probabilities for a mutation to each of the three nucleotides, given that a mutation has occurred:

$$\frac{M^g_{ij_1}}{M^g_{ij_1} + M^g_{ij_2} + M^g_{ij_3}}$$

$$\frac{M^g_{ij_2}}{M^g_{ij_1} + M^g_{ij_2} + M^g_{ij_3}}$$

$$\frac{M^g_{ij_3}}{M^g_{ij_1} + M^g_{ij_2} + M^g_{ij_3}}$$

As before, we simply need to square all these probabilities to account for them happening in two strands since the mutations in each strand are independent events. Then, we add them all together for the total probability from any scenario. This gives:

$$p = \frac{{M^g_{ij_1}}^2 + {M^g_{ij_2}}^2 + {M^g_{ij_3}}^2}{(M^g_{ij_1} + M^g_{ij_2} + M^g_{ij_3})^2}$$

as the probability of having a convergent mutation at a single site. Since this sum will be the same regardless of the initial nucleotide due to the symmetrical nature of the mutation matrix, we do not need to factor in the specific initial nucleotide; we can continue to operate generically.

With this new probability of a convergent mutation, we can use our previous Π function with a modified p value:

$$\overline{\overline{\Pi}}(c \mid \sigma) = \binom{\sigma}{c} p^c (1-p)^{\sigma - c}$$

Our model now operates under the following assumptions:

1. The number of mutations the occur on a strand is distributed according to the Poisson distribution
   (a) Mutations are independent events
   (b) Mutations are rare
   (c) The mutation rate is constant
   (d) Mutations do not occur simultaneously
2. The probabilities of each transition are equivalent
3. The probabilities of each transversion to the complementary base pair are equivalent
4. Each nucleotide site can be mutated at most once

5. The mutation rate is constant over all generations
6. Codon bias is not present
7. Selection pressure is not present

Simulation Verification

As with the previous versions of our model, we verified this modification with an accompanying simulation. This time, we use the same simulation as before with a single extra parameter:

1. g = number of generations to run the simulation for.

The simulation still outputs the total number of convergent mutations that have occurred between two strands of DNA that originated from the same ancestor, but now those mutations can occur over the course of several generations.

The simulation operates in the same manner as before with a minor modification. Now, when it comes to mutating each site along each daughter strand, it mutates each g times to simulate g generations passing. Again, the nucleotide that is in each site is based on a random weighted choice with the following weights: $1-\mu$, α, $\beta\phi$, and $\beta(1-\phi)$, so the probability still remains for no mutation to occur.

Again, we ran the simulation for 1000 iterations for various values of L and μ, and compared the running average to the expected value outputted by our model.

After confirming our model in this way, we wrote a second simulation that would allow for an even deeper analysis of our model. Our model still does not account for the phenomenon of double mutations. These occur when a single nucleotide mutates once and then mutates again in a subsequent generation. The previous simulations also did not account for this phenomenon. Thus, the results agreed with each other, but were not necessarily realistic. Thus, the goal of this simulation is to determine the relationship between mutations and convergent mutations (i.e. the ratio between the two) where double mutations are present. Then, we wanted to compare this relationship to the one predicted by our model to gain understanding about the impact of double mutations and the accuracy of our model relative to their presence. This simulation requires the following parameters from the user:

1. L = length of the DNA strands (in base pairs)
2. μ = mutation rate (in mutations per base pair per generation)
3. g = number of generations over which to run the simulation
4. κ = ratio of transitions to transversions
5. ϕ = probability of transversion to a nucleotide's complementary base pair

As in the first simulation, this one generates a random strand of DNA as the ancestor strand and replicates this strand such that we then have 2 identical daughter strands of DNA.

After generating these strands, we do not access every nucleotide as before. Instead, for each daughter, we randomly generate integers corresponding to nucleotide sites along the strand. Then, we access and mutate those sites alone. For each generation, we mutate exactly μL sites as this is the expected number of mutations that will occur in a single generation. Then, we repeat this process on each daughter strand g times to simulate multiple generations of DNA replication. Since we randomly generate sites to mutate, there is always a chance that the same site number will be generated multiple times, allowing for double mutations (or triple, quadruple, etc.). After g generations, we have placed g mutations along each strand.

After all generations, we compare the two daughter strands to each other and to the ancestor strand. This allows us to count up the number of convergent mutations between each two strands. Finally, the simulation outputs a .csv file with the data collected. The file has two variables: generations and convergent mutations. We ran this simulation for various parameter values; for each, we ran 1000 iterations and averaged the results. By generating graphs of these results we examined the relationship between convergent mutations and mutations as desired.

To check these results with our model, we also evaluated our model for generations from 1 to g.

Expected Identity Percentage

The final key aspect of our model is the expected identity percentage, which is a metric between two strands of DNA corresponding to the proportion of the genome that is the same between them. Because we can easily obtain these values from real genomic data, this provides us with a simple way to apply our model to actual bacterial genomes. From our model, we can determine the relationship between convergent mutations and ID%. Then, we can apply this relationship to the identity percentages calculated from genomic data as a method of conversion to convergent mutations. ********expand on this part once we actually do this********

Our mutation matrix can be applied over multiple generations, as explained above. From M^g, we have the probabilities of mutation to each nucleotide over the course of g generations. Thus, a single entry is the probability of a mutation from i to j (or of remaining i). From this matrix, we can obtain the expected ID% between two strands as follows.

Each M^g_{ij} value represents the probability that a nucleotide site starts with nucleotide i and ends with nucleotide j. Since each row of the matrix contains the same values just in varying orders, we can use a general i as the starting nucleotide. There are four different js that can be the ending nucleotide, so we must consider each of them. The probability that two strands of DNA end at j_1 is given by $M^g_{ij_1} * M^g_{ij_1} = {M^g_{ij_1}}^2$ because these are independent events. Likewise, the probabilities of two strands of DNA ending at j_2, j_3, and j_4 are ${M^g_{ij_2}}^2$, ${M^g_{ij_3}}^2$, and ${M^g_{ij_4}}^2$, respectively.

These are the four different ways that the two strands can be identical at a specific site; the sum will represent the total probability that the site has the same nucleotide in both strands, considering all possible scenarios; call this p_{same}. Since there are L total sites in the genome, we would expect to see $p_{same} * L$ total sites that match.

The definition of identity percentage is the number of nucleotides that are the same divided by the total number of nucleotides. Thus, we divide by L and arrive at our expected identity percentage:

$$E[ID\%] = \left(M^g_{ij_1}\right)^2 + \left(M^g_{ij_2}\right)^2 + \left(M^g_{ij_3}\right)^2 + \left(M^g_{ij_4}\right)^2$$

Simulation Verification

We then expanded the previous simulation to be used to compare with our expected identity percentage. This version of the simulation follows the same algorithm for mutating the strands as described above, but instead of outputting data about the number of mutations versus convergent mutations, it outputs a .csv file with identity percentage and convergent mutations. Identity percentage is a pairwise parameter that corresponds to the percentage of the entire genome that is the same between the strands. In other words, identity percentage is equal to the number of shared nucleotides divided by the total length of the genome. Identity percentage can also be given by the following:

$$ID\% = \frac{L - m_1^* - m_2^* + \sigma + c}{L},$$

where m_1^* and m_2^* are the number of mutation sites on strain 1 and strain 2, respectively, σ is the number of overlapping mutation sites, and c is the number of convergent mutation sites.

Again, our goal with this simulation was to determine the relationship between identity percentage and convergent mutations and compare this to what our model predicts, so we also ran our model for generations from 1 to g.

Accommodating Multi-Strand Convergent Mutations

Now, at this point our model only determined the number of convergent mutations that can be expected between two strands of DNA, but the homoplasy detector gives the number over the entire population. Thus, we need some way to generalize the value from our model to an entire species. Doing so will result in an estimation of the number of homoplasies that we expect to arise due to convergent mutation. This seems like a simple discrepancy to fix; since our model gives an estimate between a pair of strands, simply multiplying by the total number of pairs in the population ought to give us the total number of convergent present in the entire population. However, doing this will introduce inaccuracy into our estimate because of the way that homoplasies are counted. When the homoplasy detector counts up the number of homoplasies in the

population, any that happen to arise in three or more strains still counts as just one homoplasy; it counts the number of different alleles that can be defined as homoplasies. Thus, multiplying by the number of pairwise combinations could be an overestimate of the number of homoplasies caused by convergent mutations. We could see some convergent mutations that are present in three or more strands, and these should count as just one homoplasy. Thus, we cannot simply multiply by the pairwise combinations of strains.

Theoretically, we can incorporate this idea into our model using combinatorics. Employing a variant of the multinomial used to determine the probability of overlapping sites, we can redefine the probability as follows:

$$\sum_{q=3}^{n} \frac{\binom{L}{\sigma_q, m_1-\sigma_q, m_2-\sigma_q, \ldots, m_q-\sigma_q}}{\binom{L}{m_1}\binom{L}{m_2}\ldots\binom{L}{m_q}},$$

where q is the number of strains in which the convergent mutation is present. Then, using this equation as $P(\sigma \mid m_1, m_2, L)$ in our model would allow us to account for these multi-strand convergent mutations. However, this formula is computationally intensive; the multinomial coefficients involve factorials, and the values simply get too large to compute with a program. Thus, instead of working this phenomenon theoretically, we opted to use simulations and wrote another simulation to determine the expected number of convergent mutations that are present in three or more of the daughter strains.

If we return to the idea of simply multiplying $E[c]$ from our model by the number of pairwise combinations, then we can incorporate a correction factor to make up for the presence of multi-strand convergent mutations and fix with the overestimate.

This correction factor will come from estimating the number of convergent mutations that we expect to be present in three or more strains, which we will determine with simulation. Then, we can use the following equation:

$$h_c = E[c]\frac{n(n-1)}{2} - \sum_{q=3}^{n} c_q \left(\frac{q(q-1)}{2} - 1\right),$$

where h_c is the number of homoplasies due to convergent mutations, $E[c]$ is the expected number of convergent mutations predicted by our model, $n(n-1)/2$ is the well known formula for the number of pairwise combinations between n total strains of bacteria, and the summation over q is the correction factor, where q is the number of strains that display a certain convergent mutation, and c_q is the number of convergent mutations that exist in exactly q strains.

Since $E[c] * n(n-1)/2$ alone overcounts any convergent mutations that are present in more than three strands, we need to figure out how many convergent mutations we are double counting and how many times we are overcounting them.

For a convergent mutation between q strains, our model counts this once for each pair of strains. However, we only want to count it once total. Thus, our correction factor must subtract off the extras. The number of pairs that exist between q strains is given by $q(q-1)/2$, so we want to subtract all but one of these. For example, consider the case $q = 3$, then we have strains s_1, s_2, and s_3 that all have some convergent mutation. Our model will count one convergent mutation between s_1 and s_2, one between s_1 and s_3, and one between s_2 and s_3. Since, we only want it to count as one convergent mutation, so we must subtract off the extra two. Now, we must do this for each c_q, which produces the following correction factor for a convergent mutation in q:

$$c_q \left(\frac{q(q-1)}{2} - 1\right)$$

To account for convergent mutations in any number of strains over two, we simply sum this correction factor over all values of q from 3 to n. This will then give us a more accurate value for h_c, as it accounts for the phenomenon of multi-strand convergent mutations.

Simulation

Now, we need to determine what the value of c_q is. To do so, we have run simulations that mutate randomly generated strands of DNA just as the others do. This simulation requires the following parameters from the user:

1. n = number of DNA strands in the population

2. L = length of the DNA strands (in base pairs)

3. g = number of generations over which to run the simulation

4. μ = mutation rate (in mutations per base pair per generation)

5. κ = ratio of transitions to transversions

6. ϕ = probability of transversion to a nucleotide's complementary base pair

Given these parameters, it mutates n DNA strands that begin as identical copies of a common ancestor using the same algorithm as our simulation for identity percentage. Afterwards, it counts up the number of convergent mutations between each pair of strands and records the the location of each convergent mutation. Finally, it determines the number of convergent mutations that are present in 3, 4, ..., n strands. To do so, it checks each convergent mutation site and tallies up the number of strains that have the same nucleotide at that site.

Running this simulation for various parameter values, we have determined **********include specific 'critical' values**********maybe we should make graphs of c_q vs. n,L,g,etc.**********

Detecting Homoplasies

The final part of our model required detecting the number of homoplasies present in a bacterial population. To do so, we used a Python script which constructs an unrooted phylogenetic tree between the strands [30] [30]. The script characterizes the most frequent genotype at each locus as the major allele (N_0), and assigns all other alleles as minor alleles ($N_1, N_2, \ldots, N_i$) [30]. The number of nucleotide differences, D, is computed pairwise between all of the strands, and the distance data is used in the construction of the phylogeny of sequences.

In order to determine whether a given allele is homoplasic, the following metric from Bobay et. al. [30] is used:

$$\max(D(N_1N_1)) \overset{?}{\geq} \min(D(N_0N_1))$$

If the inequality holds, then the minor allele N_1 is said to be homoplasic, indicating that the allele must have arisen from either recombination or mutation. If the inequality does not hold, the minor allele is assumed to have arisen through shared descent from a common ancestor between the strands possessing the minor allele.

The program counts up all the alleles which satisfy the inequality and outputs the number, which gives us our value for the number of homoplasies.

To put everything together, we return to equation **??**: $h = r + c$. As mentioned, we estimate the number of recombinant sites, r, by simply subtracting the number of convergent mutation sites, c, predicted by our model from the number of homoplasic sites, h, outputted by the homoplasy detector.

Effects of Parameters

GC content refers to the percentage of a genome that is composed of guanine and cytosine bases. Evidence exists for both the preservation of GC content across genomes (CITE THIS) as well as a tendency for mutations to decrease the GC content overall (CITE THIS). Therefore, we wanted to test the effects of the GC composition on convergent mutations so as to determine it relevance to our model.

In order to determine the effect that genomic GC content had on the calculated rate of convergent mutation predicted by our model, we used our simulations for mutations and identity percentage. Again, these use a similar algorithm, but generating the ancestral strand changes slightly.

This version of the simulation requires one additional parameter:

1. $GC\%$ = the proportion of the ancestral genome that consists of Gs and Cs

The ancestral strand is generated such that it agrees with the given GC%. To do this, we calculated the number of Gs and Cs that needed to be present, generated those randomly, and then generated the rest of the strand by randomly attaching As and Ts. Finally, we randomly shuffled the nucleotides on the strand to

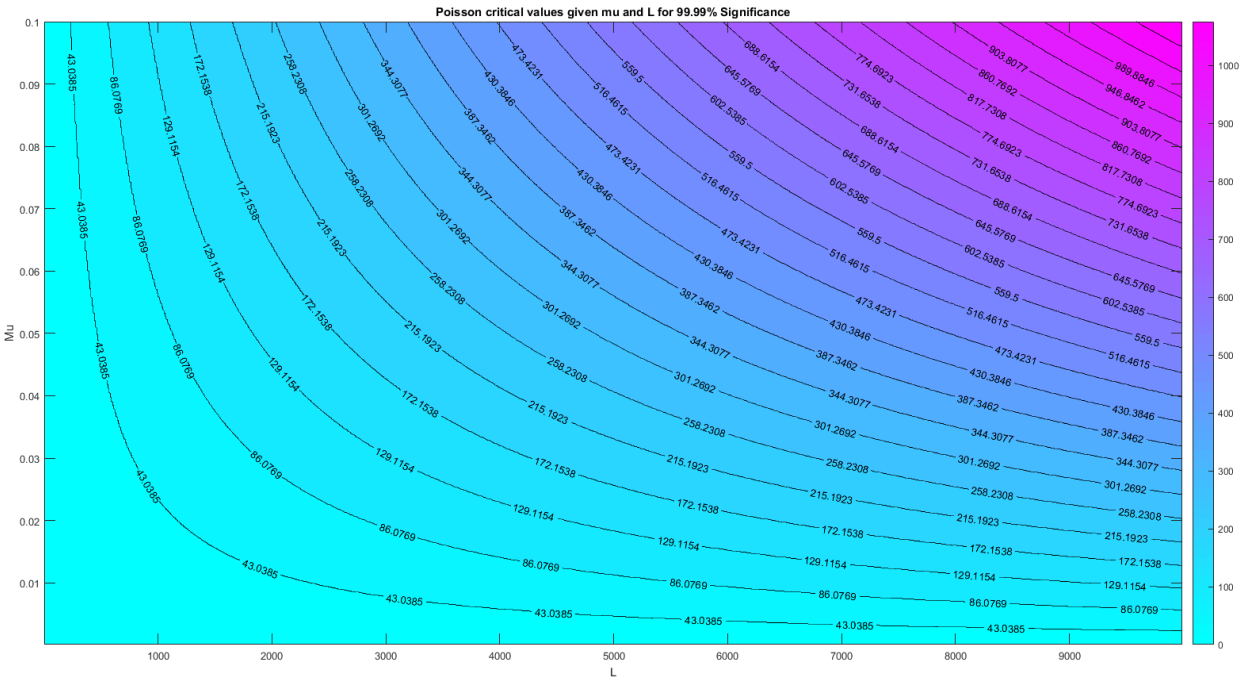


Figure 1: A contour plot of the 99.99% confidence cutoffs for various μ and L values

ensure that A, T, G, and C would be interspersed throughout. Then, as before, we replicate this strand such that we then have 2 daughter strands of DNA, and mutate them over g generations.

The simulations were run for GC percentages from 0 to 100, in increments of 10%, each with various κ values. The data for the simulation was statistically analyzed with a one-way ANOVA test, a Tukey HSD test, and a pairwise Student's T-test (with Bonferroni-adjusted p-values) to determine if GC content made a significant effect on the rate of mutation.

Current research also suggests the importance of κ in measuring mutation rates (CITE THIS). To test its impact, we ran the same simulation and the same statistical tests for various values of κ, allowing us to analyze its impact.

Adjustments for Computation

In order to have the model run efficiently on a computer, several adjustments were made to the way the calculations were performed.

First, in order to reduce the number of calculations done within each summation, the Poisson critical point which marked the α^{th} percentile of the distribution was calculated for each combination of μ and L, and the summations were indexed from zero up to this point. This resulted in $\alpha^2 \times 100\%$ of the data being included due to the multiplication of the two Poisson distributions; the excluded data was negligible with a confidence interval of $(1-\alpha)\%$. The calculated values using this factor will have a confidence level of α.

For example, consider $\mu L = (0.01)(10000)$ and $\alpha = 0.01$. In this case, the inverse Poisson function $\text{Poi}^{-1}(p = 0.99, \lambda = 100) = 124$, indicating that all of the summations can terminate at 124 rather than at $L = 10000$, and only $[1-(0.99)^2] \times 100 = 1.99\%$ of the data in the Poisson distribution is lost. For sufficiently low α the percentage of data lost will be negligible but the number of iterations per summation to complete will be significantly lower than L.

Secondly, when we programmed our model, we used logarithms to keep the magnitudes of the values in check. Because the binomial probabilities involve several factorials that can get as large as $L!$, these values can get too large to handle. Thus, working with them all in logarithmic form and then exponentiating when necessary allows us to avoid this issue. The final values of the computation are not affected by this

adjustment.

We also made a couple other computational adjustments for calculating factorials. First, we expanded the factorials and canceled out terms wherever possible. For example, the factorial

$$\binom{L}{\sigma, m_1 - \sigma, m_2 - \sigma, L - m_1 - m_2 - \sigma}$$

from equation **??** is typically evaluated by

$$\frac{L!}{\sigma!(m_1 - \sigma)!(m_2 - \sigma)!(L - m_1 - m_2 + \sigma)!}$$

However, expanding these factorials gives the following:

$$\frac{L * L - 1 * L - 2 * ... * 2 * 1}{\sigma!(m_1 - \sigma)!(m_2 - \sigma)! * L - m_1 - m_2 + \sigma * L - m_1 - m_2 + \sigma - 1 * L - m_1 - m_2 + \sigma - 2 * ... * 2 * 1},$$

which reveals that many terms cancel between the numerator and the denominator, namely, all those terms from $(L - m_1 - m_2 + \sigma)!$. This gives:

$$\frac{L * L - 1 * L - 2 * ... * L - m_1 - m_2 + \sigma + 1}{\sigma * \sigma - 1 * ... * 1 * m_1 - \sigma * m_1 - \sigma - 1 * ... * 1 * m_2 - \sigma * m_2 - \sigma - 1 * ... * 1}$$

Now, computing this fraction will involve smaller values of both the numerator and the denominator. However, we can make one more computational change: vectorization. We can vectorize both the numerator and the denominator. Since the numerator and the denominator are guaranteed to have the same number of terms, we can divide the two vectors element-wise, which produces a single vector. Then, we can multiply the elements of this vector; this will give us the exact same result as computing all the factorials would, but this method limits the size of each of the factors so that none of them are too large to handle. With this process, we are simply breaking up the computation and performing it as follows:

$$\left(\frac{L}{\sigma}\right)\left(\frac{L-1}{\sigma-1}\right)...\left(\frac{L-\sigma+1}{1}\right)\left(\frac{L-\sigma}{m_1-\sigma}\right)\left(\frac{L-\sigma-1}{m_1-\sigma-1}\right)...\left(\frac{L-m_1+1}{1}\right)\left(\frac{L-m_1}{m_2-\sigma}\right)\left(\frac{L-m_1-1}{m_2-\sigma-1}\right)...\left(\frac{1}{1}\right)$$

Testing predictions against genomic data In order to test the predictions of our model, we ran our model for the particular case of *Bacillus anthracis*. *B. anthracis* is known to be a clonal species, meaning recombination is very rare and thus any number of homoplasies detected should be attributable only to convergent mutations. The parameters of *B. anthracis* are listed below, obtained from running L.M Bobay's ConSpeciFix on the "Concatenates" data files (see Supplemental materials) [30]

For core genome:
$\kappa_{B.\ anthracis} = 1.8684$
$n = 21$
$h_{total} = 1149$
$L = 1303140$
Strains tested: {B58, B52, B51, B57, B29, B21, B4, B6, B7, B9, B34, B36, B89, B83, B84, B99, B15, B66, B64, B72, B45}

First, we calculate the average identity percentage from the data, and we use the approximation that all strains have that average identity percentage with each other.

$$ID\%_{\text{ave}} = ?????$$

Running our simulation of convergent mutations vs. ID % (Figure XXXX) for a κ value of 1.8684 and L value of 1303140, we obtain the following regression line that allows us to predict XXX convergent mutations from that

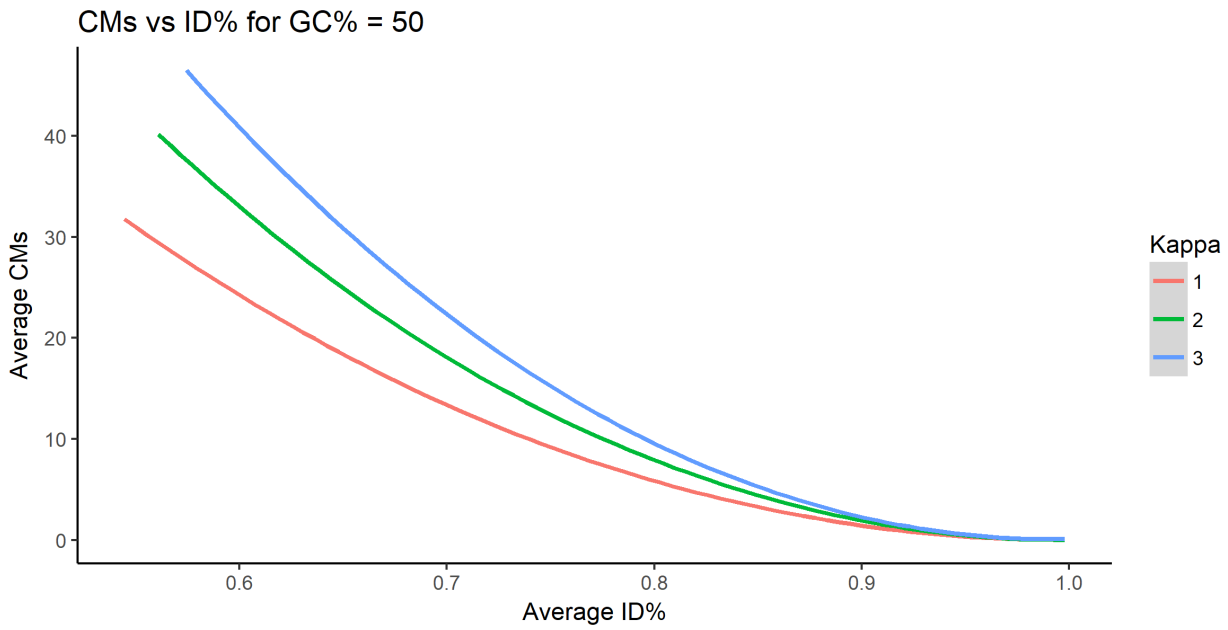


Figure 2: The total number of convergent mutations compared to the identity percentage between two strands. Trends are shown for several different values of kappa, and we note that higher kappa tends towards a higher amount of convergent mutation, especially at low identity. GC content was held constant at 50%.

SCAR Matrices Method

When analyzing genomic data, we can compare all of the strain genomes pairwise within each species. We can easily determine the amount of identical sites between genomes by simply counting the number of sites where the site number and base are both identical between the strains. Applying this method gives us a matrix where the $(i, j$ entry in the matrix is the number of identical sites between strain i and j. know that an identical site must be inherited from the ancestor, a convergent mutation, or part of a recombinant tract.

When analyzing species which have known mutation rates, we can apply our summation model over each pair of strains within a species. Applying our model to every pairwise combination gives us a matrix of the number of convergent mutations between strains.

We can also use RAxML to compute marginal ancestral reconstruction given a tree for a species. RAxML will create the maximum likelihood phylogenetic tree for a given species, and based on this tree can find the most likely ancestral sequence at each internal node of the tree.

Pyvolve Method

Results

Graphs showing the running average over each iteration of the Monte Carlo simulation are presented below. Each graph represents a different combination of parameters utilized in running the simulation, and each graph also shows a horizontal dashed line at the expected value predicted by our model.

In addition to the Monte Carlo simulations, we ran simulations to compare the number of mutations on a strand with the expected number of convergent mutations, and with identity percentage. We compared both of these datasets to our predicted values.

We attempted to separate our data by different parameters. Graphs of the data separated by kappa and GC content are shown below.

Finally, we attempted to find a relationship between the average number of convergent mutations between two strands and the identity percentage of those two strands. We fitted both a linear and quadratic model to

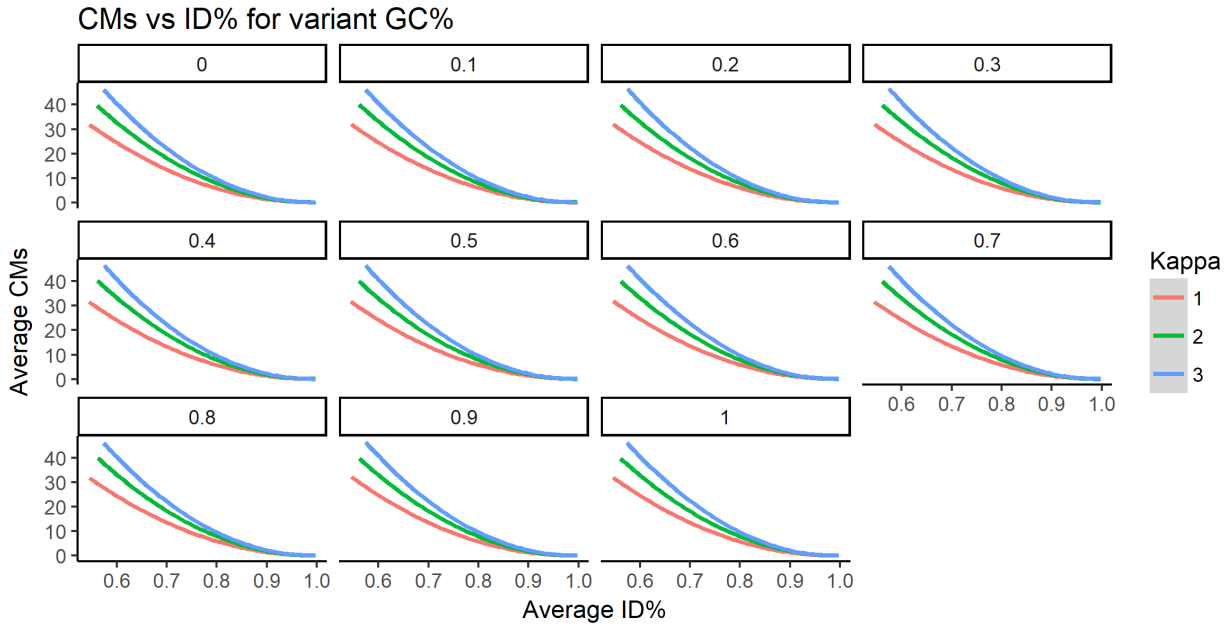


Figure 3: The total number of convergent mutations vs. ID% in one strand, grouped by kappa. Several different GC content levels were considered. An ANOVA test between convergent mutations and GC content returned a p-value of $p = 1$. To confirm these results, Tukey honest significant differences were calculated and a p-value of $p = 1$ was returned for the pairwise differences between each category.

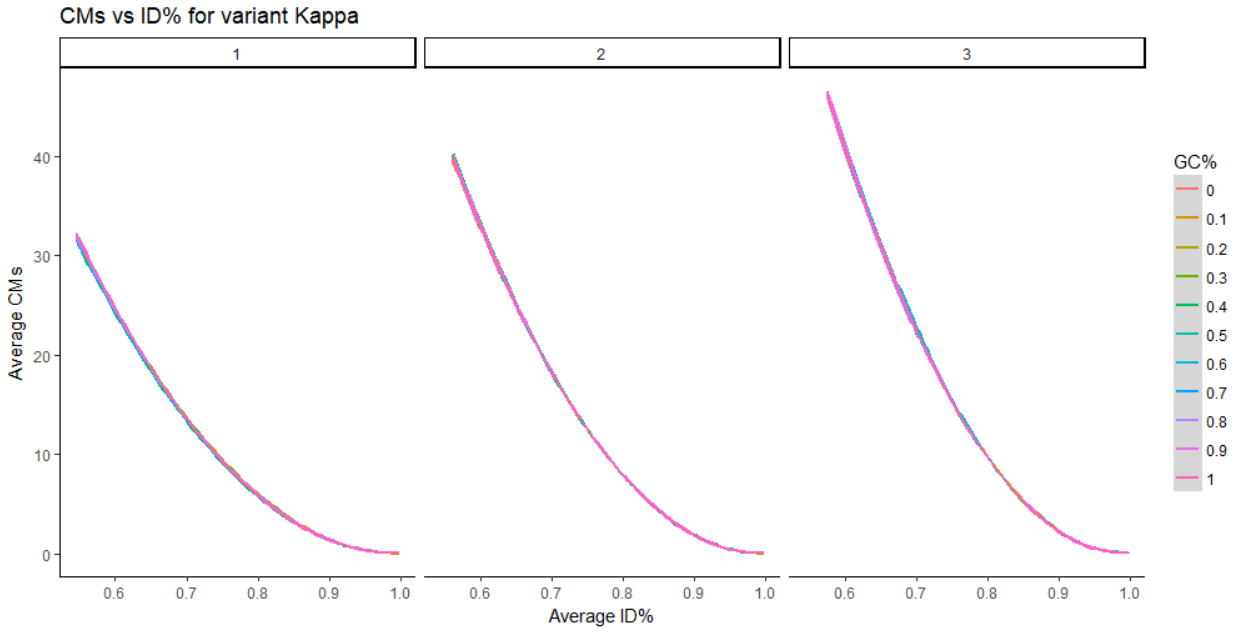


Figure 4: Several graphs of CMs vs. ID% at three different values of kappa, with overlays for each GC percentage. All of the curves for varying GC content overlap.

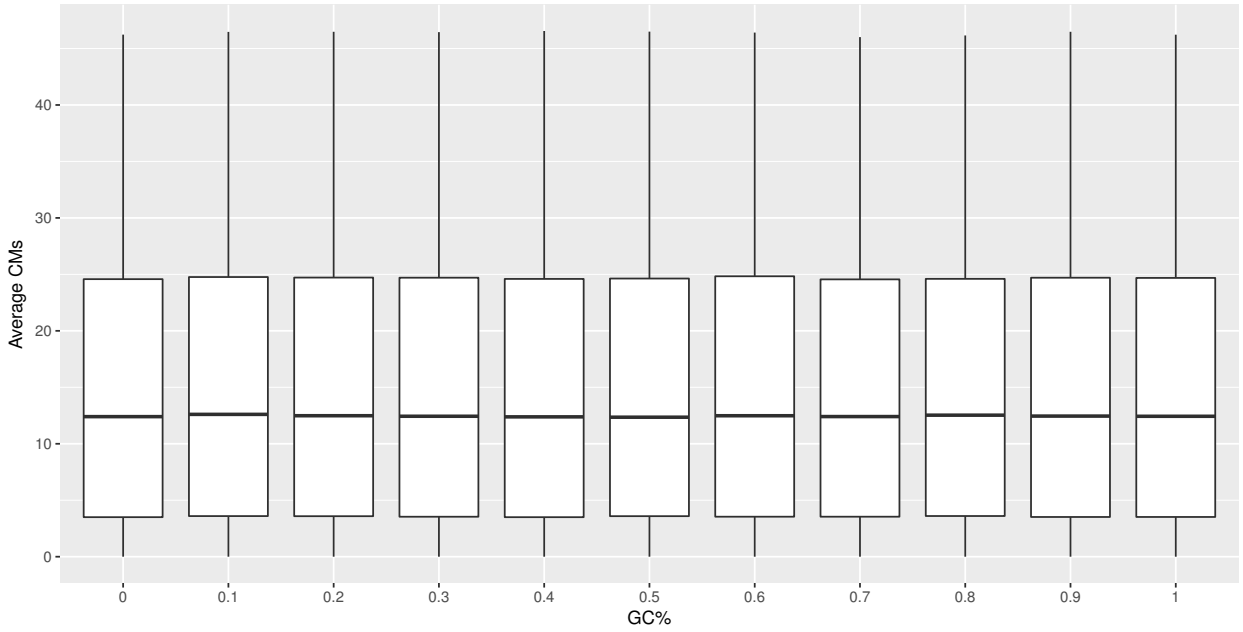


Figure 5: Boxplots for each GC percentage used. Each boxplot shows the distribution of convergent mutations for the data with each different GC percentage. A pairwise T-test with Bonferroni-adjusted p-values indicated $p = 1$ between every pair of GC distributions.

the data. The quadratic model fit with a reported $R^2 = 0.9999$, so we selected this model as the best fit for the graph. Each term in the quadratic regression line was significant. Notably, rerunning the simulation, but not allowing the occurrence of multiple mutations gave a quadratic regression line where only the a_0 term was significant. In addition, this regression line fit the data extremely well, matching up with our predicted fitting curve INSERT EQUATION FOR KAPPA CURVE HERE

INSERT NO DOUBLE MUTATION GARPH HERE

Discussion

Statistical testing to compare the different levels of GC content indicated that GC content had no effect on the rate of convergent mutations. The results of the ANOVA test suggested a lack of differences between any of the GC levels. The Tukey HSD test provided more evidence, suggesting a lack of pairwise differences between any pair of GC levels. We thus find no evidence to reject the null hypothesis that there are no significant differences between GC levels, and we conclude that the data is consistent with the null hypothesis.

Graphs

As discussed, we verified our model via several different simulations. For each stage of our model, we ran a mutation simulation and extracted data according to the Monte Carlo method. *****talk more about these once we get them in*****

Effects of Various Parameters

Here we review the effects of varying different parameters in our model.

GC Content

By running the simulations and statistical tests mentioned earlier, we demonstrate that GC content does not have any significant effect in the context of our model. This is verified through our simulations, the

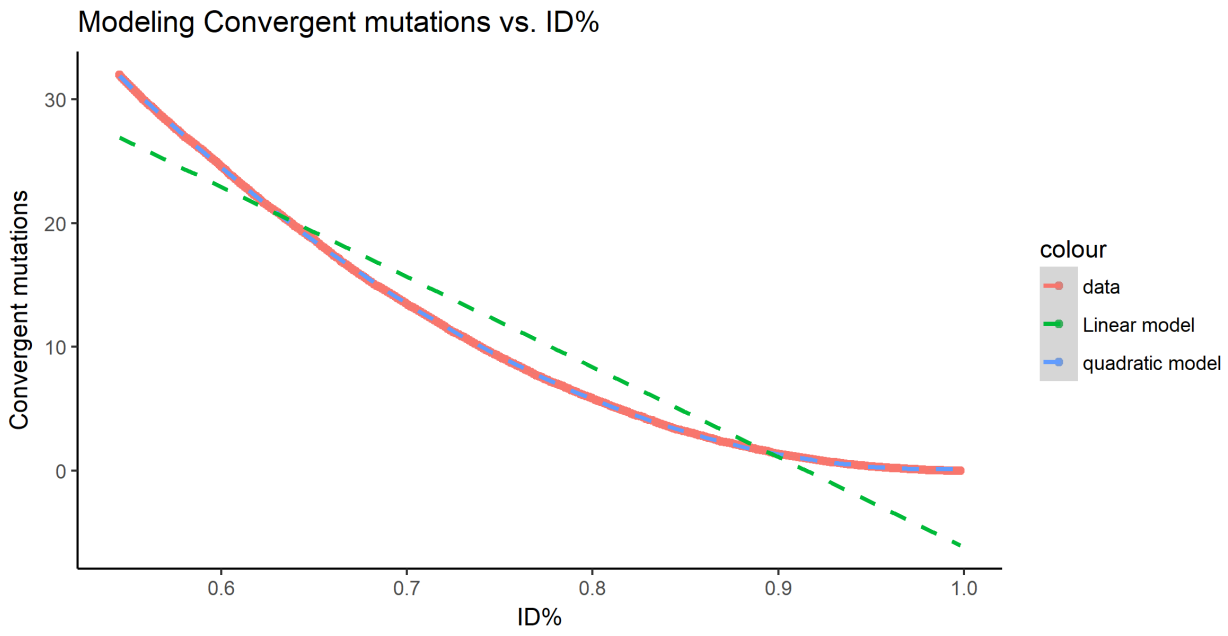


Figure 6: Two regression models, linear and quadratic, fitted to the simulated data. The linear model fit with $R^2 \approx .70$ and the quadratic with $R^2 \approx .99$. The graph shown represents GC = 50% and $\kappa = 1$, although similar results were obtained for $\kappa = \{1, 2, 3\}$

.

results of which are presented in (figure XXX), in which the effect of many GC contents are shown on a plot of convergent mutations versus total mutations. Additionally, (figure XXX) shows the impact of various GC percentages on convergent mutations versus identity percentage. The graphs suggest that GC content has little to no effect on the presence of convergent mutations with regard to our model. This inference is supported by the results of the various statistical tests performed.

With the averaged data obtained from the simulations, we used R to subset it into separate data frames by κ. One-way ANOVA was performed on the different subsets of data, with the dependent variable, number of convergent mutations, compared against the grouping factor, GC content. The p-value obtained for the one-way ANOVA was $p = 1$ for all three values of κ. To confirm these surprising results, both a Tukey HSD and pairwise Student's T-test (with Bonferroni-adjusted p-values) were performed on each of the three datasets, all of which returned similarly high p-values between every category. As suggested by these high p-values, we cannot infer a noticeable effect of GC content on the number of convergent mutations.

This lack of impact is a direct result of the parameters of our model, whereby our three parameters κ, ϕ, μ represent different probabilities for mutation that, given a matching nucleotide in the ancestral strand, lead to equivalent probabilities of convergent mutation.

This is potentially a limitation of our model, as it has been proposed that GC content may have a significant effect on the process of mutation. Bobay et. al. expresses the possibility that mutation as a whole has an AT bias, meaning that starting with a GC rich strand may lead to a different expected value of convergent mutations than an AT rich strand [29]. A potential expansion may involve adding additional parameters that distinguish between different transition and transversion rates between each nucleotide (as in the GTR model) **cite GTR model**.

Kappa

Figure XXY shows the relationship between κ and the proportion of convergent mutations expected. Here we assume $\phi = 0.5$ as this is a neutral value indicating that both transversions occur with equal probability. This shows the relationship expected with a minimum value at $\kappa = 0.5$ and increasing as kappa increases and decreases from that value. This matches intuition, as a κ of 0.5 represents an equal chance of mutating to

any of the other three nucleotides, leading to the greatest amount of diversity and thus the lowest chance of convergence.

To understand the effects of κ, we ran simulations as previous discussed and used the same statistical tests employed for analyzing the impact of GC%. Again, using R one-way ANOVA was performed on each data frame, with number of convergent mutations remaining the dependent variable, this time compared to κ. The data was not subset by GC content due to the previous results. The ANOVA resulted in a p-value less than 0.001, indicating a statistically significant difference between one or more values of κ. A Tukey HSD test, as well as pairwise Student's T-test (again, p-values were Bonferroni-adjusted) were performed to determine where the difference was. Both tests resulted in negligible p-values less than 0.001, indicating significant differences between all three values of κ tested.

This result is further confirmed in the multi-generational model, the results of which are shown in Figure XYX.

References

[1] Fred Griffith. The significance of *Pneumococcal* types. *J. Hyg. (Lond.)*, 27:113–159, 1928.

[2] Oswald T. Avery, Colin M. MacLeod, and Maclyn McCarty. Studies on the chemical nature of the substance inducing transformation of *Pneuomococcal* types. *J. Exp. Med.*, 79:137–158, 1944.

[3] Michael G. Lorenz and Wilfried Wackernagel. Bacterial gene transfer by natural genetic transformation in the environment. *Microbiol. Rev.*, 58:563–602, 1994.

[4] Genetic exchange in *Salmonella*. *J. Bacteriol.*, 64:679–699, 1952.

[5] Anthony J.F. Griffiths, Jeffery H. Miller, David T. Suzuki, Richard C. Lewontin, and William M. Gelbert. *An Introduction to Genetic Analysis*. W.H. Freeman, New York, 7 edition, 2000.

[6] Gareth A. Cromie, John C. Connelly, and David R.F. Leach. Recombination at double-strand breaks and dna ends: conserved mechanisms from phage to humans. *Mol. Cell*, 8:1163–1174, 2011.

[7] Li Xuan and Wolf-Dietrich Heyer. Homologous recombination in DNA repair and DNA damage tolerance. *Cell Res.*, 18:99–113, 2008.

[8] Micheal M. Cox. Recombinational DNA repair in bacteria and the RecA protein. *Prog. Nucleic Acid Res. Mol. Biol.*, 63:311–366, 1999.

[9] Zhucheng Chen, Haijuan Yang, and Nikola P. Pavletich. Mechanism of homologous recombination from the reca–ssdna/dsdna structures. *Nature*, 453:489–494, 2008.

[10] Howard H.Y. Chang, Nicholas R. Pannuzio, Noritak Adachi, and Michael R. Lieber. Non-homologous DNA end joining and alternative pathways to double-strand break repair. *Nat. Rev. Mol. Cell Biol.*, 18:495–506, 2017.

[11] Frits Ørskov and Ida Ørskov. Summary of a workshop on the clone concept in the epidemiology, taxonomy, and evolution of the enterobacteriaceae and other bacteria. *J. Infect. Dis.*, 148:346–357, 1983.

[12] John Maynard Smith, Noel H. Smith, Maria O'Rourke, and Brian G. Spratt. How clonal are bacteria? *Proc. Natl. Acad. Sci. USA*, 90:4384–4388, 1993.

[13] Michiel Vos and Xavier Didelot. A comparison of homologous recombination rates in bacteria and archaea. *The ISME Journal*, 3:199–208, 2009.

[14] Joseph Hermann Muller. The relation of recombination to mutational advance. *Mutat. Res. Fund. Mol. Mech. Mut.*, 1:2–9, 1964.

[15] Joseph Felsenstein. The evolutionary advantage of recombination. *Genetics*, 78:737–756, 1974.

[16] Bürger Reinhardt. Evolution of genetic variability and the advantage of sex and recombination in changing environments. *Genetics*, 153:1055–1069, 1999.

[17] Leigh Van Valen. A new evolutionary law. *Evol. Theory*, 1:1–30, 1973.

[18] Claus O. Wilke. The speed of adaptation in large asexual populations. *Genetics*, 167:2045–2053, 2004.

[19] Phillip Gerrish. The rhythm of microbial adaptation. *Nature*, 413:299–302, 2001.

[20] Phillip Gerrish and Richard E. Lenski. The fate of competing beneficial mutations in an asexual population. *Genetica*, 102, 1998.

[21] William Hanage. Not so simple after all: Bacteria, their population genetics, and recombination. *Cold Spring Harbor Perspect. Biol.*, 2016.

[22] Stanley Sawyer. Statistical tests for detecting gene conversion. *Mol. Biol. Evol.*, 6:526–538, 1989.

[23] John Maynard Smith. Analyzing the mosaic structure of genes. *J. Mol. Evol.*, 34:126–129, 1992.

[24] John Maynard Smith and Noel H. Smith. Detecting recombination from gene trees. *Mol. Biol. Evol.*, 15:590–599, 1998.

[25] Daniel Posada. Evaluation of methods for detecting recombination from dna sequences: Empirical data. *Mol. Evol. Biol.*, 19:708–717, 2002.

[26] Xavier Didelot and Daniel Falush. Inference of bacterial microevolution using multilocus sequence data. *Genetics*, 175:1251—-1266, 2007.

[27] Michael P.H. Stumpf and Gilean A.T. McVean. Estimating recombination rates from population-genetic data. *Nat. Rev. Genet.*, 4:959–968, 2003.

[28] Louis-Marie Bobay and Howard Ochman. Biological species are universal across life's domains. *genome biol. evol.*, 9:491–501, 2017.

[29] Louis-Marie Bobay and Howard Ochman. Impact of recombination on the base composition of bacteria and archaea. *mol. bio. evol.*, 34:2627–2636, 2017.

[30] Louis-Marie Bobay, Brian S. Ellis, and Howard Ochman. ConSpeciFix: classifying prokaryotic species. *Bioinformatics*, 2018.