UNIVERSITY OF PRETORIA

Mathematical simulations and Verhulst modeling of compositional changes in DNA sequences of acquired genomic islands due to bacterial genome amelioration

Master Thesis

Xiaoyu Yu

11/13/2014

Submitted in partial fulfilment of the degree: Msc Bioinformatics, Bioinformatics and Computational Biology Unit, Department of Biochemistry School of Biological Sciences, Faculty Natural and Agricultural Science University of Pretoria

Table of Contents

Declaration

I, **Xiaoyu Yu** declare that the thesis, which I hereby submit for the degree **Msc Bioinformatics** at the University of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

SIGNATURE: ……………………………..
DATE: ………………………………….

Literature Review

1) Horizontal Gene Transfer

1.1) Background and Process

Horizontal gene transfer (HGT), as the name states, is a process that transfers genetic material from one organism to another by genealogical reproductive way (also known as vertical transfer of genetic material). The transfer can occur between the same organisms or across different species but mainly within the prokaryotic species and seldom in eukaryotes. There are at least three mechanisms when it comes to the process of HGT. Conjugation, where mobile elements such as plasmids actively replicates itself and gets transferred to another cell; transduction, where the gene of a host cell is packaged within a virus and transferred to another host along with virus genes and transformation is when parts of the DNA are picked up from external environment (Figure 1).

The first sign of HGT was in an experiment by Joshua Lederberg and Edward L. Tatum which saw a type of bacterial mating called conjugation. The experiment observed that the generation of daughter cells is able to grow in a media that cannot support the growth of either of the parent cells. Their experiments showed that this type of gene exchange requires direct contact between bacteria (Lederberg and Tatum 1946). Later in the early 1950s, following the success of the study where genetic exchange happens between the bacteria *Escherichia coli*, the authors hypothesized further that all bacteria could undergo such a process and hence experimenting on *Salmonella typhimurium* and other *Salmonella* serotypes began. The results from the experiment were positive and later on the process were named transduction forms one of the three main mechanisms of HGT. (Zinder and Lederberg, 1962).

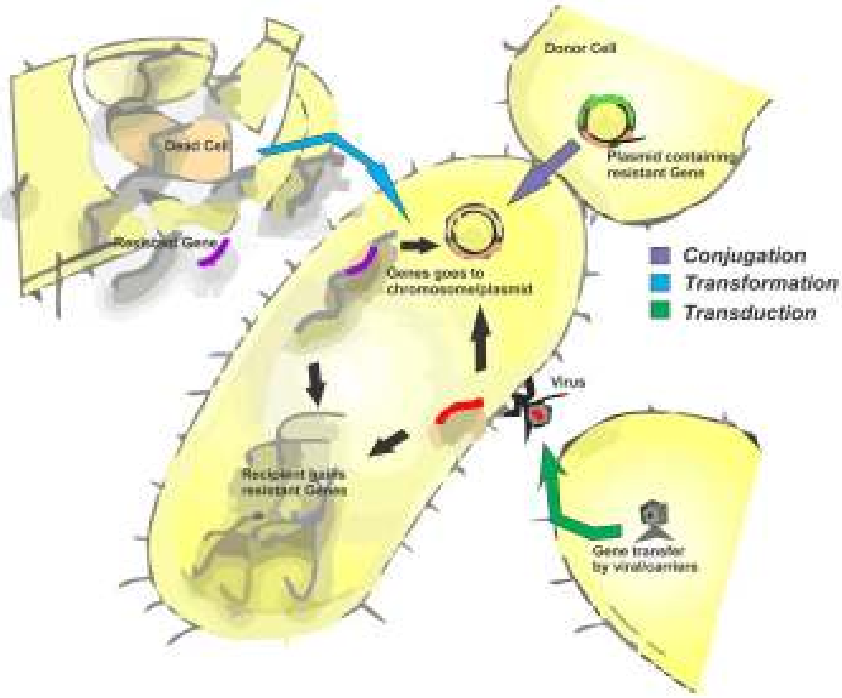

Donor Cell
Dead Cell
Plasmid containing resistant Gene
Resistant Gene
Genes goes to chromosome/plasmid
■ Conjugation
■ Transformation
■ Transduction
Virus
Recipient gains resistant Genes
Gene transfer by viral/carriers

Fig 1. The process of HGT can happen in three different ways. Conjugation, where mobile element such as plasmid replicates itself and transferred to the recipient; transformation happens when DNA material is picked up from external environments (dead cells) and transduction occurs when genes are packaged in viruses and transferred when changing hosts.

In the mid of 1980s, the phenomenon and impact of HGT was first described by Michael Syvanen. The hypothesis was that cross-species gene transfer in prokaryotes occurs and a new perspective in prokaryote evolution could be considered. The idea was that the uniform genetic code across different species allows exploitation of the same transferred gene by different organisms. There are also some other hints that genes can be transferred between prokaryotes, hints such as DNA containing plasmid which autonomously replicate, transfer by phages such as the experiment by Freeman or by direct digestion, phagocytosis. By ingesting large amount of genetic material, the bacteria which are main carrion eaters are exposed to genes from most other species. All these cases are ways which leads to HGT and also the building blocks of how we know the concept it is today. The major point in Syvanen's article was that lateral gene transfer is a major concept in order to explain many things that the classical Darwin theory cannot explain in evolution and also contributes many new possibilities into the way of evolution (Syvanen, 1985).

1.2) Advantages and Disadvantages

Standing in the bacteria point of view, there are two major advantages to HGT to be considered. The first advantage would be gaining of a new function that is beneficial to the recipient through transfer of a novel gene from a foreign donor. This will rapidly increase the rate in which the species will evolve compared to evolving independently. Another benefit is when an organism undergoes gene loss by deletion or deleterious mutations and will be able to regain that gene through HGT by another member of the population.

Of course on the other side of the coin, there are also many disadvantages that are very detrimental to the host. When HGT occurs, the transferred genetic material is somewhat random (Complexity Theory discussed later on in the evolution section) and also the point of insertion is also random. Hence majority of the time, HGT is not beneficial and one or more of the following can occur. Non-coding genes could lengthen the size of the genome and hence increase the replication time of RNA/DNA. Genes with no function (could be due to other interactive genes not being present within recipient) or duplicated genes could be expressed increasing translation and transcription cost which is not beneficial to the host. Random insertion of genes could lead to existing genes being none functional or interfere with the gene function. Hence HGT is a high risk high reward process.

With numerous advantages and disadvantages, finding the balance of how much HGT is actually beneficial for optimal evolution rate is then important. In a study done by Higgs and Vogan, they modeled the beneficial and detrimental effects of HGT (Vogan and Higgs, 2011). By testing different values of HGT rate and gene loss rate within the model, they were able to conclude that HGT rate was high when gene loss was high (due to advantage two) and vice versa. They further hypothesized that the earliest genomes before the last universal common ancestor had high gene loss during replication process and hence HGT was favored. As the genes are rapidly spread, larger and fitter genomes were built, vertical transfer of these genomes can then be passed down with lower gene loss rate. This can be seen by modern prokaryotic genomes which have a much lower HGT rate since the chance of a beneficial gene being transferred is relatively low compare to earlier genomes. Therefore increases the probability of detrimental effects by HGT and hence lower HGT rate is preferred.

When we look at the traditional portrait of the tree of life, we could see that HGT is not something that happens frequently. This is due to the fact that with the increasing number of complete genome sequences being available, we could see that some genes are highly conserved but slightly discordant. If the rate of HGT is high, then majority of the genome sequences from different organisms should be similar and all organisms should function in the same way. An explanation to this could be that when the genome sequence of an organism reaches a particular level of complexity, gene transfer should not be possible since the organism already obtain the gene or the gene is not necessary to the organism. This way, the rate of HGT will decrease and reach the Darwinian Threshold (the time of major transition of evolutionary mechanisms from mostly horizontal to mostly vertical transfer). In this case, HGT is a disadvantage and the rate is minimized in order to achieve better evolution which matches the conclusion that Higgs and Vogan came up with the model.

1.3) Genomic Islands

With the increasing importance of HGT within the bacterial world, we needed to find a way in which we can detect the actual transferred genetic region in order to get a better understanding of the effects of HGT. Genomic Islands (GI), a region of genetic material that is foreign within the host organism that is thought to be transferred over by HGT. The idea came from pathogenic islands (PAI) which as the name states is a region of genetic material that can cause the bacteria to become pathogenic when it was not before. The initial naming of PAI was by Groisman and Ochman in 1996 where studies showed viruses transferred virulent strains from one to another and hence come out of a dormant state after a certain amount of time. PAIs were then characterized as an unstable region with virulence-associated phenotype (Groisman and Ochman, 1996). Some other GI types include symbiosis (Sullivan et al., 2002), metabolic (Penn et al. 2009), antibiotic synthesis and antibiotic resistance (Levings et al. 2005) and fitness (Hacker and Carniel, 2001). These GIs vary in size and recognized by their functional homology. GIs are normally between 10 to 200kb and could be recognized by compositional means such as GC content, GC skewness, tetranucleotide and/or codons frequency biases. Phylogenetic methods can also be used to find characteristics such as having 16-20bp direct repeats flanking on both sides allowing integration of GIs into target site. Some other characteristics such as GIs containing cryptic genes encoding integrases and carrying of insertion elements or transposons (Buchrieser et al., 1998; Gal-Mor and Finlay, 2006) could be present. These characteristics are all markers for identifying GIs in target sequences.

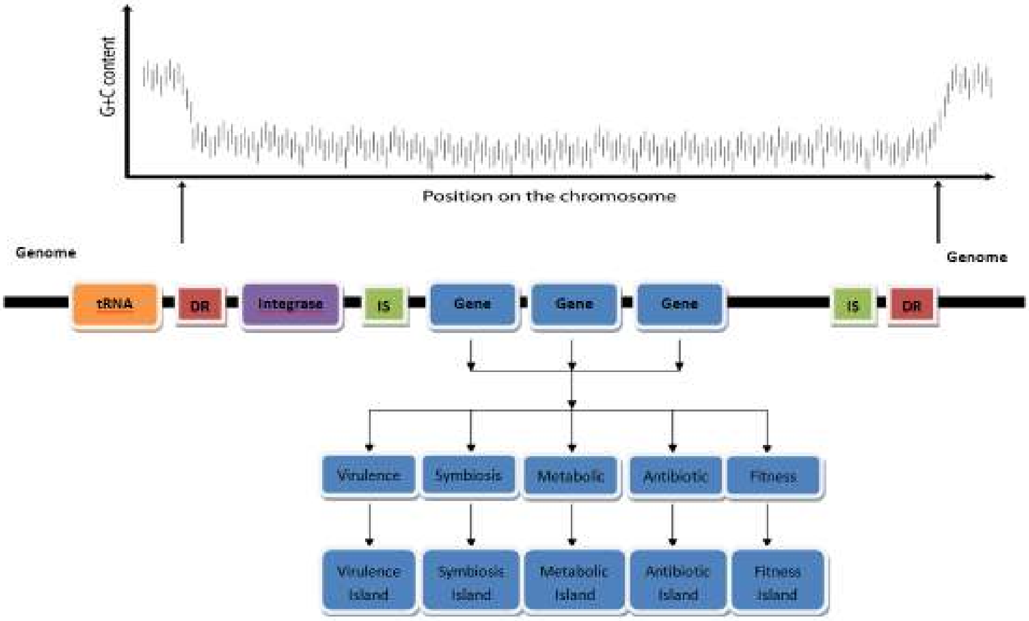


Fig 2. The general structure of a genomic island which contains irregular GC content compared to the rest of the genome. Genomic islands are usually inserted after tRNA and is flanked both sides with direct repeats (DR). It also contains an integrase as well as some insertion sequence and some genes. The type of genomic island will be determined according to the type of gene. Genomic Island can be described as virulence, symbiosis, metabolic, antibiotic and fitness islands.

With current technology, HGT have never been an easy task to identify. There have been many methods developed but none have had perfect success in identifying all HGT events. Each method has its own pros and cons and no real golden standard in how to really identify HGT. However when a GI is located, further studies could be done in order to investigate the evolutionary effects of HGT in a retrospective manner by performing function annotation and analyzing the amelioration of inserted DNA regions within bacterial genomes (HGT).

1.4) Evolution

Ever since the discovery of the concept of HGT until modern understanding of the process, it is evident that HGT is one of the more important driving forces in the world of prokaryote evolution. From a recent study by Babic *et al.*, they were able to get a direct visualization of horizontal gene transfer. Using *E. coli* as the test subject and fluorescent protein fusion method, they found that DNA transfer through the F pilus at considerable cell distances and the integrated transfer DNA through recombination occurred within up to 96% recipients. These transferred DNA also split and segregate to different chromosomes through successive replication and future generations inherit different cell clusters (Babic et al., 2008). From the above example, the process could be explained as a section of history within an organism evolutionary timeline.

HGT has been a rather controversial topic when it comes to explaining some parts of evolution but it is widely accepted in prokaryotic world (Boucher et al., 2003). Unlike natural selection, mutation and genetic drift which are some other evolution processes, when lateral gene transfer happens across species, the genes transferred are not always appropriate genes for the recipient genome. With large amount of genetic material being transferred, could all these be useful to the organism which receives all these genetic material? While, inappropriate gene transfers may lead to destruction of organisms since random insertion of genetic material could lead to a change in many protein functions. Novel functional gene transfer however will improve adaptability and survivability of the organism through many new functions gained by the transfer. Since the organism with the transferred gained novel function which improve survivability, by natural selection, this organism will outlive the others and hence pass down the newly combined genetic material to the next generation. Hence, HGT also fits in the explanation of evolution.

There have also been problems regarding the lateral gene transfer model in terms of evolution. The main concern was that by looking at the genetic material in different organism, one can hardly tell which gene has been transferred and which organism was the donor. Today with numerous whole genome sequences being available, many criteria have been set to check where such transfers have occurred. One such criterion is based on codon biases and bases compositions relative to the genes in the DNA sequence, but this criterion itself is very controversial due to the fact that the mutation rate or selective pressure in the recipient is different from the donor (Koski et al., 2001). Another problem in identifying HGT is that the DNA sequence itself mutates over time and that the DNA composition of transferred genes ameliorates and become more like the genes within the recipient. This makes it hard for researchers to identify the donor of the gene and make a connection to HGT even happening. By the above criterion, identification of gene transfer also becomes harder when the transfer happened a long time ago. Another major concern to HGT is that when comparing genes, one cannot tell if the gene has been lost or being transferred.

As the technology in this field of study increases exponentially, so does the amount of research put into HGT in order to get a clearer picture of the effect on evolution. Results showed that 1.6 to 32.6 percent, depending on the microbial genome, may have been acquired by horizontal gene transfer (Koonin et al., 2001) and recently using network analysis of shared genes, the above result could be increased to around 81 percent (Dagan et al., 2008). This is due to the fact that current HGT identifying technique cannot identify all HGT events and different techniques also give different results with little overlap. Current techniques also have high false positive rates hence the estimate of HGT genes within different microbial genomes have a large range in percentage. With the above results, current microbial genomes share a large percentage (up to 81 percent) of its genome between them and hence rare appropriate genes have a less probability of getting transferred. This would lead to less HGT rate which matches the case of modern microbial genomes.

In another study by Kanhere and Vingron (2009), it was shown that transfer across specie in the microbial world happens more often from bacteria to archaea (from the study, 74% of genes were transferred in this direction) instead of the other way around. Out of the transferred genes from bacteria to archaea, majority of the transferred genes were closely related to metabolic functions. On the other hand, archaea gene transfers to bacteria showed a preference of translational related genes(Kanhere and Vingron, 2009).

Acknowledging that HGT is an important factor of bacterial evolution, it should be accounted that there are certain barriers in which HGT is limited to a certain extent. The complexity theory proposed by Jain *et al.* contains two points. First point says that the informational genes such as genes involved in DNA replication, transcription and translation are less prone to HGT while operational genes are more likely to be transferred (Jain et al., 1999). This point was supported by Nakamura where he published an article using Bayesian inferences showing that operational genes were more likely to be transferred (Nakamura et al., 2004). The second point in the complexity theory was that after the genes have been transferred, post-transfer maintenance of genes occurs and the genes with useful functions are preserved while useless genes were removed. In this case, the organism will rapidly gain new functions. Another aspect to be considered is the effect of taxonomic distance between organisms which would effect if HGT would occur or not. Several studies shows that gene transfers occur more effectively if the two organisms participating in the transfer are closely related in terms of evolutionary (Nakamura et al., 2004, Ochman et al., 2000).

Building onto the complexity theory, Wellner *et al.* proposed that when an organism achieved a certain complexity, it serves as a barrier to prevent HGT (Wellner et al., 2007). They further hypothesized that connectivity (gene interaction network to form protein complex) is also associated with HGT where gene with lower connectivity has a higher chance to be transferred. This makes sense since a new genes with lesser connectivity is easier to incorporate into a genome. HGT is beneficial when it introduces a new gene to the recipient genome to create new function but when the genome itself is complex enough, the transferred gene more likely will cause harm on native genes or fail with incorporation into an existing network of genes. Hence a lower HGT rate is observed between taxonomically distant organisms which are caused by taxonomic barrier. This idea is also supported by Vogan and Higgs (2011) in their model where HGT reaches a certain threshold in which vertical transfer of genes is more beneficial than horizontally (This phenomenon is also known as Darwinian Threshold). Study by Mozhayskiy and Tagkopoulos (2011) also confirm the above points with another measure of fitness where environmental complexity will also affect the rate of horizontal gene transfer.

The tree of life for prokaryote species currently is difficult to define. With numerous cross transfers of genetic material, speciation itself is a complex process hence new ways for speciation is needed (Thompson, 2013). The uses of phylogenetic trees are no longer a viable way to explain complex relationships between species although there are still new methods still in development to define such systems (Thiergart et al., 2014). Phylogenetic tree (rooted or unrooted) which takes into consideration the most parsimonious (MP) connection (represented by branches, caused by speciation or mutation) between most common ancestor and species (represented by nodes) after speciation event. This linear model is restricting by the fact that most of the time there are more than one MP connection between species and creating a single phylogenetic tree from data is often contradicting and lots of results of interest could be lost depending on the type of research being done (phylogenetic tree only takes into consideration the MP connection, hence other possible MP connections will be lost and not analyzed).

This brings us to a more generalized model of phylogenetic tree also known as a phylogenetic network which considers more than just a single MP connection but multiple possible MP connections between species under study (Husan and Bryant, 2006). This technique is less limiting and hence broadening up the ability to do more research of interest such as complex relationships between bacteria species or characteristics of different networks under different evolutionary circumstances. The downside to all this is of course the computational cost of taking multiple MP connection into account (as many MP connections as possible whereas taking all MP connection into consideration is sometimes impossible with current technology). There are different types of phylogenetic networks depending on the research done (Figure 3). Split networks, a more generalized phylogenetic tree which takes into consideration multiple MP connections between species and their most common ancestor into one super tree. Reticulate networks display evolutionary data with events such as hybridization, recombination and HGT which fits very well with bacteria. Other types of phylogenetic networks also exist. These include gene loss and duplication as well as host and parasite co-evolution. Aside from MP type analysis, other analysis methods also exist such as statistical parsimony (Templeton et al., 1992).

Pangenomics (bacterial species can be described by its pan-genome, which is composed of a "core genome" and a "dispensable genome") became a more viable way to explain species of bacteria (Medini et al., 2005*)*. The core genome of bacterial specie could be considered as the household genes and dispensable genome are the genes that have either gone through HGT or through mutation. Based on the idea of phylogenetic networks, core genome can also be seen as sharing of a common ancestral history while dispensable genes can be the reticulate events such as HGT which are the multiple branches within the network. With so many genes transferred and a large pool of dispensable genomes, identifying HGT events becomes increasingly more difficult.

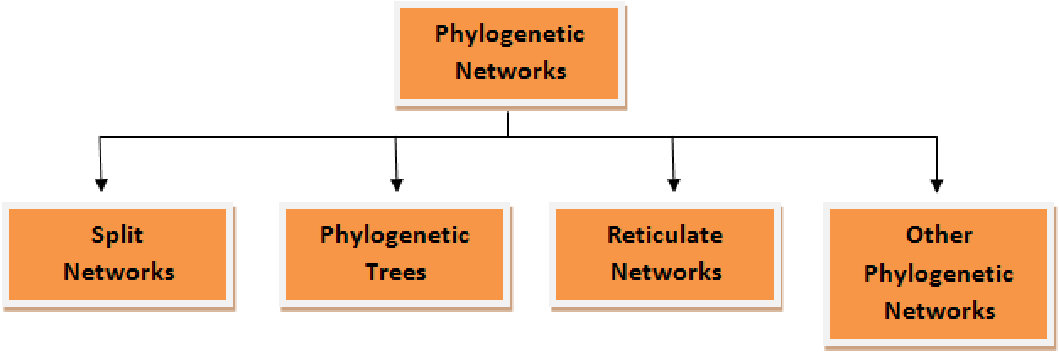


Fig 3. Different types of Phylogenetic networks, phylogenetic tree being one type of network. Other networks include split networks, reticulate networks and other phylogenetic networks. Each type of network is used for different type of research depending on research. Split network include Median network (data from sequence), Consensus network (data from tree), split decomposition and neighbor-net (both data from distances). Reticulate network include hybridization network (data from tree), recombination network (data from sequence) and ancestral recombination graphs (data from genealogies). Lastly, other phylogenetic network include any graphs explaining evolutionary data and augmented trees where HGT is represented as additional inserted edges into the tree to create a network of not just linear transfer but also horizontal.

2) HGT Identification tools

2.1) Phylogenetic Approach

When it comes to identifying HGT, there is no single bioinformatic tool capable of finding all HGT within an entire genome. Currently phylogenetic and compositional methods are the two main groups to be considered with the most success. While each of the above groups has their own strength and weaknesses, the results from both of these approaches barely overlap and hence hard to distinguish which method is better or more correct (Figure 4). Phylogenetic method searches for conflicts between the phylogeny inferred for a gene and the assumed organismal phylogeny whereas compositional methods searches for atypical regions within a genome compared to the rest of the genome. The reason for the non overlapping set of results from both methods is amelioration whereby transferred genes will undergo directional mutation and hence harder to detect using compositional methods. Compositional methods detect largely recent events which depends on donor and recipient having different compositional traits, while phylogenetic methods depend on homolog sequence being present in other sequences separating donor and recipient which allows this methods to detect much more ancient transfers (Ragan et al., 2006).

The first step to any phylogenetic method is to collect a large amount of data sequence to infer trees for the comparison analysis. A big downfall for this method is that sometimes when there is insufficient phylogenetic information, a lot of HGT events cannot be detected. Suppose that the data set is sufficient and a phylogenetic tree is built based on the sequences (normally using ribosomal RNA or well conserved and characterized protein sequences (Santos and Ochman, 2004)), a reference tree is also made based on the true evolutionary history of the organisms under study. The second drawback happens during tree building phase whereby both trees are based on many trees built and come to a consensus tree. Due to different rates of mutation of different genes, this process could be challenging and a consensus tree could be hard to determine. Taking the drawback into account, after both trees have been built, a comparison is done between the two trees and if HGT event has occurred within this data set, there should be a disagreement between the two trees. There are many ways to do a tree comparison, but the optimal measure is using the subtree prune and regraft (SPR) distance (Hein et al., 1996) to find HGT events within a tree.

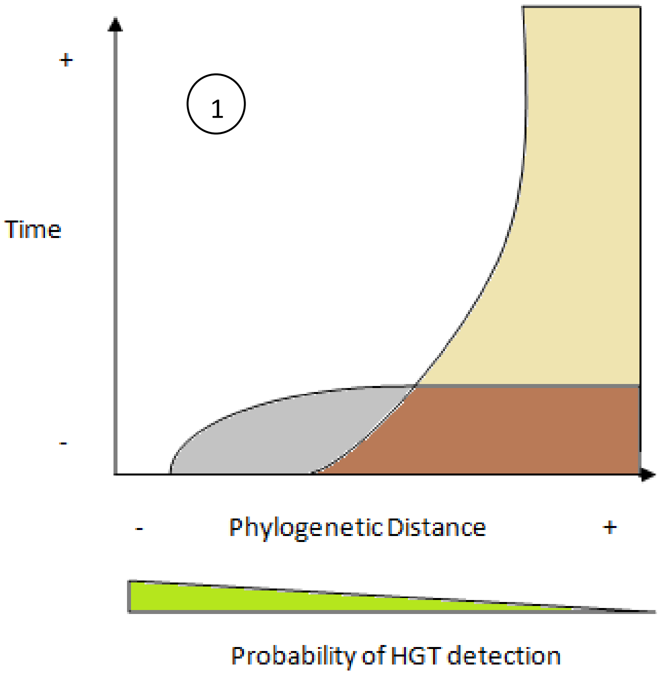


Fig. 4. Graph indicating the detection of HGT by compositional and phylogenetic methods in terms of time and phylogenetic distance parameters. Since amelioration occurs at a faster rate than normal mutations through vertical transfer, older HGT events are detected by mainly phylogenetic methods (Beige) while compositional methods detect more HGT events since HGT happens more frequently between close taxa (Grey) which are one of the limitations of phylogenetic method. There are HGT events identified by both methods (brown) which the conditions needed to identify the process are satisfied. The region indicated by (1) where HGT happened long ago between close taxa is difficult to identify through either methods hence currently none of the existing methods can identify all HGT events within a genome.

SPR operation on a tree is defined by performing a cutting on any edge hence pruning subtree “t” and then regrafting “t” to a new vortex (Figure 5). In the context of HGT, the regrafted edge corresponds to the donor and the cut edge corresponds to the recipient. An edit path is a set of SPR operations which was done to the reference tree in order to get a congruent tree compared to the inferred one. Normally there will be many edit paths for a single comparison and hence an optimal edit path is chosen which is the most parsimonious for a given reference and inferred tree. The length of the optimal edit path is then the minimum SPR distance between the trees which also include all HGT events within this data set. One advantage of the phylogenetic approach is that when doing the analysis, one can see the direction of transfer during the HGT event based on the optimal edit path (Beiko and Hamilton, 2006). While this approach is very powerful in detecting HGT events, computational complexity in order calculate SPR is still limited.

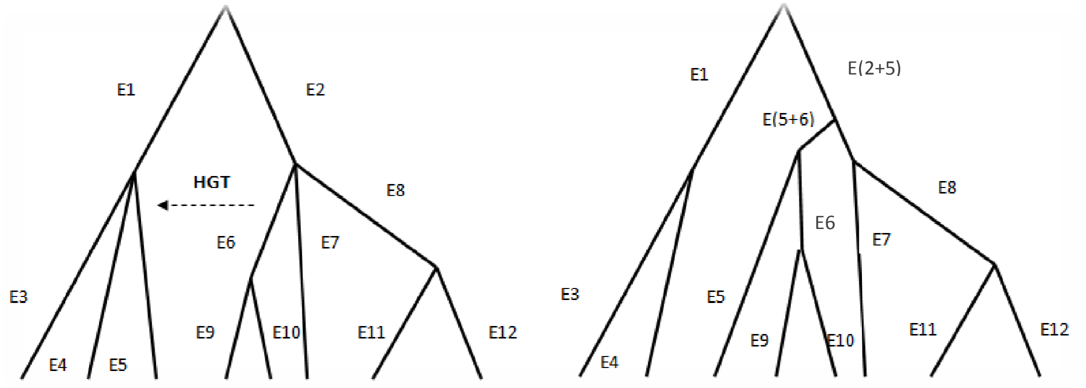


Fig. 5. SPR operation is done when there is a discrepancy between the reference tree and the inferred tree. Suppose HGT occurred indicated by the dashed arrow, edge E6 is then the donor and is received by edge E5. SPR operation start by cutting edge E5 and then regrafting it under the new vortex E(5+6). New parent edges are formed by incorporating the new edge to create E(2+5). The other edges are not affected by the operation since they were not involved in the transfer event.

Despite many challenges faced by phylogenetic methods, it is still the method of choice in terms of identifying HGT especially for ancient genes. Aside from the SPR distance, there are other statistical tests that boost the power of accuracy on phylogenetic methods. The approximately biased (AU) test is such a measure whereby for each tree tested, a probability is calculated for the confidence that this tree is the true tree describing history of the data under consideration. The greater the P-value for the test tree, the closer it is to the true tree and a confidence set is made containing all test trees with a P-value above a significant alpha value. If the confidence set does not agree with the organismal phylogeny with significant alpha value, then there are possible HGT event within the data set. The AU test is reliable in its measure with respect to false positives but false negative rates are still within unacceptable range especially with stringent alpha values. Depending on research, a decrease in alpha value could be used to decrease false negative rates. A 5% significant level gives up to 90% power of detection on average which is reasonable but still inadequate in some cases.

Building onto the traditional method of phylogenetics, a method of genome wide prediction of HGT can be done in retrospective assessment of prediction reliability. With current technology, genomic information such as annotation and new genome sequences are rapidly increasing and therefore an automated and computational efficient tool should be created. With the addition of the above two points, the downside of phylogenetic methods greatly decreases and therefore a stronger tool can be created. Darkhorse algorithm created by Podell and Gaasterland combines a probability based lineage weighted selection method with filtering approach and adjustable for wide variation in protein sequences conservation to detect HGT on a genome bases (Podell and Gaasterland, 2007). The algorithm uses an unique measure namely lineage probability index (LPI) which is calculated by using BLAST results in the relative context (based on the query by a percentage base to compensate difference in conservation between proteins) and then ranked by their lineage frequency of matches over entire genome (based on multiple databases which increases statistical power). The measure can characterize organism's HGT history profile, density of database covered for related species and list of proteins least likely to be inherited. The algorithm is made to be efficient (only LPI measure is needed to be computed) and automated, therefore useful in quick updates of incorporating new information to existing analysis. Positive results can then be prioritized for more in depth analysis such as phylogenetic tree and nucleotide composition.

2.2) Compositional Approach

Compositional method, also known as parametric method, uses characteristics of the genome as a tool in detecting HGT events. These characteristics include GC content (as well as first and third codon position), oligonucleotide usage (OU), and codon bias *etc*. These parameters all have its pros and cons and all had its successes in detecting HGT. GC content being the most basic method of composition group uses the theory of each specie has its own unique GC content pattern. This is due to a combination of environmental (adaptation and survivability in different habitat) and genetic factor of individual genome (Foerstner et al., 2005; Sueoka, 1988). Hence finding atypical regions of highly differential GC content within recipient genome allows identification of HGT events.

Since GC content methods are the most basic, there are many flaws and needs many improvements. Nucleotide positions are known to mutate at different rates depending on their region within the genome (conservative regions) or different codon positions (third codon positions tend to mutate more than the other two). Therefore depending on the GC methods alone will create many false positives and will not be a sufficient method to use in order to detect HGT. Codon usage is the next addition to the tool box which reinforces the existing GC method and gives solution to the above two problems. Aside from taking into consideration of comparing the GC content of both first and third codon positions from the genome mean, codon adaption index (CAI) is also used as a measure to differentiate atypical genes within recipient (Sharp and Li 1987). A statistical chi squared test is used to give power to the test in order to reduce amount of false positives.

Although with the addition of codon usage technique, the detection power of the above two method increased significantly. Codon bias and GC content is still a poor indicator for HGT and still many transferred genes are still under the detection radar (Koski et al., 2001). Oligonucleotide usage (OU), also known as short nucleotide sequences or k-mer (sequence length of two to fourteen nucleotides), are known for its descriptive characteristic of a genome. OU signature dates back to 1995 where Karlin *et al.* uses dinucleotide composition bias to make evolutionary implications (Karlin and Burge, 1995). Since then, statistical approaches were used as a reinforcement of existing OU techniques (Deschavanne et al 1999) as well as higher order k-mer analysis such as tetranucleotide patterns along with Markov Models (Pride 2003). The pattern of deviation of OU frequencies from expectation (where combination of ACGT of the same length had the same frequency) were shown to be genomic signatures and hence contain phylogenetic characteristics that links microorganisms. Therefore the idea behind this approach is that genomic OU composition within genome is less variable than between genomes. This allows simplistic criteria in order to identify HGT events and region (GI) regardless of where the analysis within genome being considered which codon bias technique lack in. OU statistics also take into consideration interactions between nucleotides such as base stacking energy, position preference and bendability which can affect the rate in which the nucleotide mutate (Reva ON 2004).

Like the other techniques, OU pattern analysis in terms of detecting HGT is powerful but still lacking from being the perfect approach. An overview done by Bohlin *et al.* which analyzed the effectiveness of di-, tetra- and hexa-mers in detecting HGT came to an conclusion that none of the above three techniques were superior than the others and are all context dependent (Bohlin et al., 2008). Thus lack of a golden standard in choosing which approach to use in different context to prevent false positive and false negative predictions is a key problem of GI identification. Therefore, new algorithms need to be made by incorporating as much information as possible as well as being flexible and simplistic to use. SeqWord Genomic Island Sniffer (SWGIS), software developed by Bezuidt et al. (2009) uses OU statistics to identify atypical regions within genomes. OU pattern of various length could be used in combinations of each other (di- to hepta-nucleotides) within one algorithm which gives flexibility and an all round analysis to identify HGT (Reva, 2005). SWGIS belongs to the larger SeqWord project by Reva which also contain other software tools such as genome browser where GIs can be visualized and differentiated into different types (Ganesan et al., 2008).

SWGIS utilizes three parameters in order to detect and differentiate different GIs. These parameters include OU distance, pattern skew (PS) and OU variance. Normalization could also be done which allows frequency count of words to be normalized by other word length or mono-nucleotide frequencies (e.g. if the analysis is done on a much skewed or diverse GC content genome, you would like to normalize the word frequency by mono-nucleotide to take that fact into consideration in your analysis). Normalization is split into two options, internal and external which internal applies to the current genomic fragment under analysis and external applies to the global genome.

The program first calculates the frequency of nucleotide words of various lengths (chosen by user) and deviation of frequency from expected is then calculated and recorded as a matrix (pattern). The results are then ranked according to most deviated to the least deviated. Distance between two patterns (e.g. GI region compared to the whole genome) is calculated as the absolute distance between ranks of oligonucleotide word in the two patterns. The program automatically calculates four combinations of direct and reverse strands and takes the minimum value as the distance. Therefore depending on the distance value, HGT events can be identified since genomic signature is somewhat unique to each organism, a large distance value between patterns shows that there is a foreign genetic material. Pattern skew is a particular case of distance measure which calculates the distance between direct and inverse strands of the same DNA. Since for bacterial genomes the PS value tends to be low, a high PS value could imply insertion of phage elements (Reva, 2004). Lastly OU variance calculates the variance of the deviation between two patterns. Depending on the normalization used on the patterns, the patterns are unique and a large difference in variance between patterns is another criterion for identifying HGT events. However, due to there being a constraint to the number of combinations of nucleotide words, uncontrolled mutation (insertion) can cause higher OUV values which could cause false positives which is a downside to this algorithm.

There are other applications of compositional methods beside the standard analysis of whole genome sequences. Tamames and Moya developed an algorithm which estimates the extent of HGT in metagenomics (Tamames and Moya, 2008). Since compositional methods require comparison of region of interest to rest of the genome, metagenomic data is therefore lacking in this regard. An alternative approach is then used by combining OU (tetranucleotide used here) by sliding windows (10 at default) through ORF and Pearson's correlation between the windows. ORFs are compared to each other and low correlation implies dissimilarity between them. All values are then grouped into matrix and then clustered into a tree. Depending on the cutoff, any significant correlation values (lower than cutoff) are then considered to be transferred genes. Though the task is difficult in identifying HGT with metagenomic sequences, the results are still a good start into a new field of study. In turn, compositional techniques are a powerful tool in many applications and a simplistic yet efficient way in identifying HGT events.

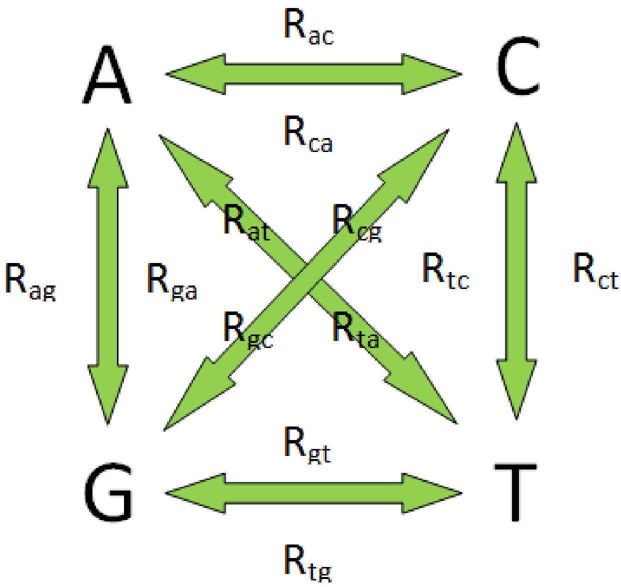
2.3) Other Approaches

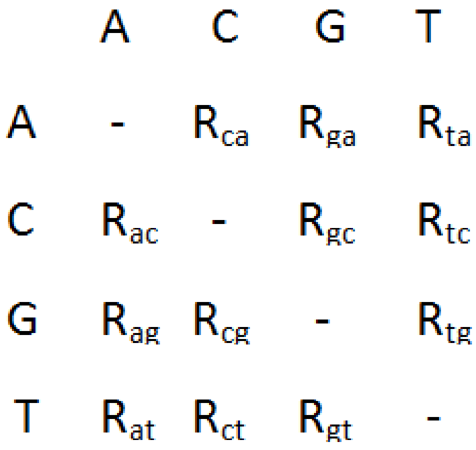
New techniques are being brought up at incredible speed with modern technology. Aside from the standard phylogenetic and compositional approaches to identify HGT, other methods that utilize similar ideas to the above two main groups have emerged. We here look at two of these techniques and see what new perspective these results will bring and their shortcomings compared to the other main techniques.

The theory behind the first technique branches off from traditional compositional group whereby using the nucleotide substitution rate matrix to detect HGT. Different species have their own nucleotide compositions and hence must have their unique rate matrix associated with it. This offers an advantage over traditional compositional methods whereby similar composition does not imply similar rate matrix. Hamady *et al.* hypothesized that HGT changes nucleotide substitution dynamics because mutational processes differ between old and new organism (Hamady et al., 2006). Hence if a change in the rate matrix is detected, HGT should occur within the organism since the transferred gene rate matrix differ from the recipient and by amelioration, a change of rate matrix occur. On the other hand if a genome has not undergone HGT, the rate matrix should stay moderately the same between genes of the same organism. A criterion is then set for a test for putative HGT genes within different genomes.

The rate matrix is derived from the Markov Model of neutral sequence evolution. This algorithm is used because of its success in many other bioinformatic applications such as sequence searching, alignment and phylogeny. The model typically represents four nucleotides at any given position within a DNA sequence. Each nucleotide has a rate of change to other nucleotide and is then grouped as a 4x4 matrix with each unit within the matrix representing a rate from one nucleotide to another (Figure 5a). By theory, the row of the rate matrix must sum to zero because the rate of change away from each state must equal the rate of change towards them. Hence the diagonal values must be negative while the off diagonal values are positive (Figure 5b). This is one of the criteria to check if the rate matrix is correctly derived. The rate matrix for the genome is then derived empirically through the probability matrix through a logarithmic conversion. Although Markov Model is useful, many assumptions for the model do not always make biological sense. Assumptions such as all sites are identical and independent are clearly not true in the sense that sites are often correlated especially those that encode RNA (Smith et al., 2004). Markov models also have a basic assumption of time consistency and being time-reversible which is also not biologically correct (Lobry and Lobry, 1999). To get close to being biologically significant, a constraint must be added to the rate matrix. This in turn limits the true inference of the rate matrix and therefore shrinking the ability to detect HGT which is the major downfall of this approach.

For this model specifically, two improvements have been added in order to improve biological significance while not decrease the accuracy of the model in determining HGT. Triple roots are used instead of pairs to remove the assumption of time-reversibility. This allows further increase in accuracy in order to determine the rate matrix as well as allow direction of change to be inferred at the cost of a bit more computational time. The model also only takes into consideration the nucleotide of the third codon position to minimize the influence of selection.





	A	C	G	T
A	-	R_{ca}	R_{ga}	R_{ta}
C	R_{ac}	-	R_{gc}	R_{tc}
G	R_{ag}	R_{cg}	-	R_{tg}
T	R_{at}	R_{ct}	R_{gt}	-

Fig. 5a. Rate of change between different nucleotide states by Markov Model.
Fig. 5b. Rate of change matrix based on the Markov Model. The negative diagonal values allows the rows to sum to zero which is a criteria for a valid rate matrix

In order to check the extent in which the ability of the model to detect HGT is significant, inferring the rate matrix correctly is of great importance. The rate matrix is inferred using three genomes of the same species of approximately the same level of divergence. If the rate matrix inferred does not fit well with the genes within each genome (uncorrelated) in some way, this implies that there should be more than one rate matrix that fits this genome and hence HGT even should have occurred. To discriminate if there are two or more rate matrices involved, phylogenetic methods are used to build trees based on the number of rate matrices involved of 8 or 16 sequences. Various new statistics are then used in order to analyze the trees to make a conclusion. These statistics include a combination of different methods used by other researchers which include mean distance of sequence, number of sequence used for inferring and normalizing of the rate matrix, variance of distance etc.

This method of detecting HGT increases accuracy in discriminating between HGT and non HGT events up to three folds compared to standard GC content methods. The comparison to GC methods is due to it being consistent within bacterial genomes as well as all other compositional statistics are somewhat related to GC content. This improvement of accuracy is based on a combination of different statistics in order to reduce error rate. The random forest algorithm was used in order to include statistical significance to the result (Breiman, 2001). There is a limit to the accuracy of this method whereby the rate matrix between the compared organisms must be slightly different in order to distinguish a difference. While some downfall is still evident, overall, this new approach offers a new tool to detect HGT at significant accuracy.

The second technique differ from the first in which it uses pure statistical analysis in order to detect HGT but still branches off the standard compositional approach. Chatterjee *et al.* proposed that at any segment of a whole chromosomal sequence must have a similar distance between the segment and the rest of the sequence (Chatterjee et al., 2008). The measure could be done based on GC content or their oligonucleotide distribution and the distance measure are done by either absolute or Euclidean distance. Alternatively for annotated genomes, one may take account gene content and their codon usage as well as amino acid usage biases.

The first phase of the test for HGT is done by a comparison of “s” to the rest of the chromosome sequence (“s” being a segment within the DNA sequence under study). A vector of N segments is taken independently and each segment has the same length as “s” and does not overlap with “s”. Another vector of N random pairs is then independently selected from the chromosome sequence which does not overlap with “s”. Distance is then calculated for both vectors between “s” and s' (complement of “s”) as well as s_1' and s_2'. If s belongs to a GI, vector D_1 should be larger than D_2 otherwise HGT did not occur in the segment s.

Statistical theory states that since D_1 and D_2 are both taken independently hence both vectors are independent and identical distributions. Therefore a standard statistical test can be done whereby the null hypothesis is that elements in D_1 is the same as element in D_2 (not a GI) and alternative hypothesis would be D_1 is larger than D_2 (GI). Mean and variance values are then calculated for both vectors D_1 and D_2 and by central limit theory (for large enough N) if the statistic is zero, null hypothesis is true while if the statistic is positive, alternative hypothesis is true. If computational constraint is an issue, a smaller N must be chosen and therefore a two sample Kolmogorov-Smirnov test or Wilcoxon-Mann-Whitney statistical test can be used as a replacement (Randles, 1979). These tests support the above criteria as a test for GI.

Since GI vary in size and location within the chromosome, statistical test is then done on s by sliding window across the chromosome and varies in size. The sliding window should not be too small as it will increase the computational cost significantly. P-value is then recorded and plotted allows easier detection of GI location and size within chromosome. A cutoff P_0 ($0 < P_0 < 1$) then determines the putative GI from the chromosome which ends phase one of the tests. Further refinement is done on the putative GIs after phase one since these GIs are always larger in size and may be a false positive. A refinement phase is then done to increase the accuracy of the method by reducing false positive. This phase is similar to the above test but the only difference is that the putative islands are removed from the chromosome sequence itself so that the random segment does not include any of these regions. This will in turn reduce the effects of any influence by any putative islands present in the chromosome sequence. The rest of the test will be the same as the first phase.

The method works well and ranked highest compared to some other well known methods such as Island-DB, W8 and HGT-DB etc. in terms of sensitivity but not so well in terms of specificity. This is due to the fact that this method detects a much larger number of putative GIs based on the criteria used. The advantage of this method is that it does not require any training set and uses a powerful statistical backup which some other methods lack. On the other hand, possible downfalls of this method include computational cost with varying window sizes and a high number of false positives which is caused by a large number of putative GIs being identified. While setting strict parameters can reduce the false positive rate, a lot of interesting information could be lost so a clear border line is hard to distinguish.

3) Amelioration Model

3.1) Concept of Genomic Amelioration

Amelioration is the process where the base DNA composition of the transferred genes from a donor undergoes nucleotide substitutions over time and reflects similarly in DNA composition to the recipient genome. This is due to the fact that the introduced genes are subjected to the same mutational pressure (Sueoka, 1988) as the recipient genome and hence over time become more similar in genomic composition. This is directed mutation whereby the foreign insert within a new environment being under the stress of a new mutational pressure and hence undergoes increased selection towards the recipient genome. Amelioration can hence be thought as an evolutionary process or model whereby acquiring foreign genetic material and making it its own for its own benefits through stress induced mutagenesis is also viable (Maclean, 2013). The process of amelioration is more evident in large groups of gene transfers since there is a larger region of atypical composition to undergo directional mutation. Furthermore, newly transferred genes are easier to identify since they have just started ameliorating and comparison of donor and transferred gene can aid the modeling process.

There have been very few models of amelioration since the start of this idea and therefore there is no golden standard to the approach. The most famous one is to use the rate of nucleotide substitution between the gene transferred and recipient genome to model amelioration (Lawrence and Ochman, 1997). The rate and extent of amelioration as well as analyzing how long each gene undergoes directional mutational pressure allows the estimation of the time of HGT. This model is based upon the fact that nucleotide composition of a DNA sequence typically represents an equilibrium between selection and directional mutational pressure (Sueoka, 1962; Sueoka, 1988). When a gene is transferred, the gene will experience the same directional mutational pressure as the recipient genome and its base composition will reach a new equilibrium. Mathematical model describing this change has been developed (Sueoka, 1962) to express this change in DNA composition with respect to directional mutational pressure. The amelioration model does not directly quantify directional mutation pressure but rather represent it as a fraction of net change in DNA composition with regards to nucleotide substitution rate. The model consists of four parameters namely nucleotide substitution rate, transition/transversion rate (IV ratio), GC content at equilibrium and GC content of HGT region. All parameters are easily calculated and the model itself is very easy to use.

The rate of amelioration can be expressed as a function in terms of the substitution rate. An empirical substitution rate S can be expressed as the rate of change at the site of cytosine or guanine (R_{GC}) and the rate of change at site adenine and thymine (R_{AT}). R_{GC} can be further broken down into the rate of change from G or C to A or T ($R_{GC \text{->} AT}$ includes G -> A, G -> T, C -> A and C -> T) and the rate of interchange between G to C ($R_{GC \text{->} CG}$ includes G -> C and C -> G), similarly for R_{AT}. Knowing the fact that all transition mutation and half the transversion changes the GC content of the DNA sequence, we can simplify the equation into one rate of change along with the IV ratio. One assumption for this is that the two transversion rates are equally frequent.

$$S \;= [(IV\ Ratio + 1) / (IV\ Ratio + 0.5)] \times [R_{GC \text{->} AT} + R_{AT \text{->} GC}] \qquad [1]$$

Equation [1] represents the total substitution rate in terms of both directional mutations and transition/transversion rates. The combined action of these two rates will lead to GC equilibrium as proposed by Sueoka (Sueoka, 1962, Sueoka, 1988). GC equilibrium (GC_{EQ}) can therefore be expressed as a ratio between the directional mutation rate of AT and GC.

$$GC_{EQ} = R_{AT \text{->} GC} / (R_{AT \text{->} GC} + R_{GC \text{->} AT}) \text{ and } AT_{EQ} = R_{GC \text{->} AT} / (R_{AT \text{->} GC} + R_{GC \text{->} AT}) \qquad [2]$$

Therefore combining [1] and [2]

$$R_{AT \text{->} GC} = S \times GC_{EQ} \times [(IV\ Ratio + 0.5) / (IV\ Ratio + 1)] \qquad [3a]$$

$$R_{GC \text{->} AT} = S \times AT_{EQ} \times [(IV\ Ratio + 0.5) / (IV\ Ratio + 1)] \qquad [3b]$$

The GC change over time can therefore be expressed as the gain in GC content minus the loss in GC content. Let AT_{HGT} and GC_{HGT} be the base composition of the horizontal transferred DNA:

$$\Delta\ GC_{HGT} = [AT_{HGT} \times R_{AT \text{->} GC}] - [GC_{HGT} \times R_{GC \text{->} AT}] \qquad [4]$$

Combining (3a), (3b) and (4):

$$\Delta\ GC_{HGT} = S \times [(IV\ Ratio + 0.5) / (IV\ Ratio + 1)] \times [GC_{EQ} - GC_{HGT}] \qquad [5]$$

Equation [5] shows that the rate of change in GC content of horizontal transferred DNA can be expressed by three parameters. Δ GC_{HGT} is proportional to the substitution rate (S) as well as the difference between the GC equilibrium and the GC content of the horizontal transferred DNA values. The above two parameters as well as the IV ratio can all be derived from comparative studies in nucleotide sequences. Also taking into consideration that different codon positions experience different selective pressure and mutate at different rates. Hence different codon positions can be analyzed independently to create a more accurate amelioration model for specific DNA sequence.

Even with taking into consideration that different codon positions experience different mutation rate, the amelioration model proposed above is still too simplistic. Taking the whole region into consideration, using a single mutation rate (S) might be insufficient to explain amelioration process. But the three parameters within the equation is still sufficient to make biological sense for the amelioration model to work, but a lot of information can still be added to improve the accuracy of the existing model.

3.2) Project Aims

Based on the model described in the previous section, the use of single nucleotide substitution rate S and GC content difference between equilibrium and transferred region takes the most basic assumptions for it to make biological sense. Due to its simplicity, it is easy to use at the cost of much information lost in result. Similarly to the GC content method in compositional techniques, taking in consideration of only single nucleotide within analysis causes flaws. These flaws can be covered by using oligonucleotide patterns which takes many of the assumptions into account which single nucleotide analysis lacks in. Hence a model which uses oligonucleotide pattern data needs to be derived for a more detailed analysis on amelioration of bacterial genomes.

OU statistics calculated by the SWGIS are a very useful tool which summarizes important characteristics of a genome. By taking into consideration the difference between the OU statistics of GI and the rest of the donor genome allows us to get a clearer picture the amelioration process. Amelioration is the process in which the foreign genomic material will undergo mutation to achieve similar composition to the recipient genomic sequence. Hence the OU pattern of the GI will tend towards the donor sequence therefore difference in OU distance and variance will decrease during the amelioration process.

Using this fact, conversion can done on the OU word distance between GI and recipient sequence into a probability of mutation per iteration and utilizing these parameters to create a simulation of the amelioration process.

Verhulst model is well known for its uses within the biology field for modeling population growth (Horowitz et al., 2010; Koseki and Nonaka, 2012). The sigmoid curve which defines the model is useful in explaining the amelioration process (Exponential increase in the beginning as well as the decreasing towards capacity value trending towards the end shows directional mutation which is the core assumption of the amelioration process) as well as its simplicity to use (Simple logistic equation). The parameters of the model also reflect real life situations (capacity, initial exponential increase then decrease towards capacity). Therefore we aim to model the amelioration process through the usage of Verhulst Model and oligonucleotide usage patterns within the sequences of genomic island and recipient.

3.3) Project Objectives

The objective of the project will be to derive an algorithm which will model genomic amelioration of bacterial genome using a combination of compositional methods (OU) and mathematical modeling (Verhulst Equation). The algorithm should reflect the dynamics of the amelioration process and produce parameters that could explain it throughout the time lapse of the process. These parameters must be biological meaningful in which it can explain the trend of the amelioration process as well as estimate the time of insertion of different horizontally transferred genomic islands in different recipient genomes. Hence multiple different genomic islands (testers) as well as possible recipient genomes (target) were chosen such that the algorithm can be seen in terms of explaining amelioration for all types of bacteria genome sequences. These different tester target combinations should also be able to give answers questions regarding the amelioration process. These include different taxa amelioration comparisons (gram-positive, proteobacteria, *etc*.), from the same taxa (is the amelioration process less extreme than different taxa?) and tester target sequence from the same genome (does it ameliorate at average mutation rate/ no directional mutation?).

The resulting model must be simplistic and easy to derive in the sense that by using the least amount of core data and using most autonomous method to achieve. The model should also be able to make conclusions such as rate of mutation estimate of different genomic loci sequences and the time of insertion of different genomic island into their recipients. To ensure the most accurate results, we use multiple genomic islands from the same origin to compare the degree of amelioration to their time of insertion.

Vocabulary

GI – Genomic Island

OU – Oligonucleotide Usage

PLF – Probability Logistic Function

PAGI - *Pseudomonas aeruginosa* Genomic Island

Tester – Genomic Island insert sequence

Target – Recipient genome sequence

Introduction

Horizontal gene transfer (HGT) within bacteria studies has dated back several decades and has been well documented. Current studies are still undergoing to dwell deeper into its effects within phylogeny and evolution alongside improvement in new technology and techniques (Hamady et al., 2006). These techniques have been improved to increase accuracy in determining HGT events as well as trying to create a standardized tool which can determine all HGT events. Currently there are two main methods in determining HGT, compositional and phylogenetic, in which both has their own advantages and disadvantages (Vogan and Higgs, 2011). Amelioration, the process where the base DNA composition of the transferred genes from a donor undergoes nucleotide substitutions over time and reflects similarly in DNA composition to the recipient genome, is one of the factors influencing the creation of a standardize tool and a major downfall of compositional methods.

Although HGT is well understood, amelioration itself is understudied. Hence the study of amelioration is vital to enhance the understanding of this process. With this insight, many aspects such as the mutation process of transferred material (preference in composition mutation, directional mutation, mutation rate), and the effects of base composition of recipient on the amelioration process can be answered. Therefore we attempt to create a logical yet practical mathematical model to model amelioration.

To increase the understanding of amelioration within bacterial genomes, four foreign inserts and known genomic islands (GI) were used to model their amelioration process towards compositional profiles of genomes of organisms representing distant taxa and different GC content, i.e. Bacillus subtilis 168, Pseudomonas aeruginosa PA01, Escherichia coli K12, Xylella fastidiosa 9a5c and Streptomyces griseus NBRC 13350. These genome sequences were chosen as a small sample in attempt to cover the vast bacteria domain with many major phyla being covered (actinobacteria, gram-positive bacteria, *etc*.). Hence the amelioration model created here can also be applied on other bacteria organisms.

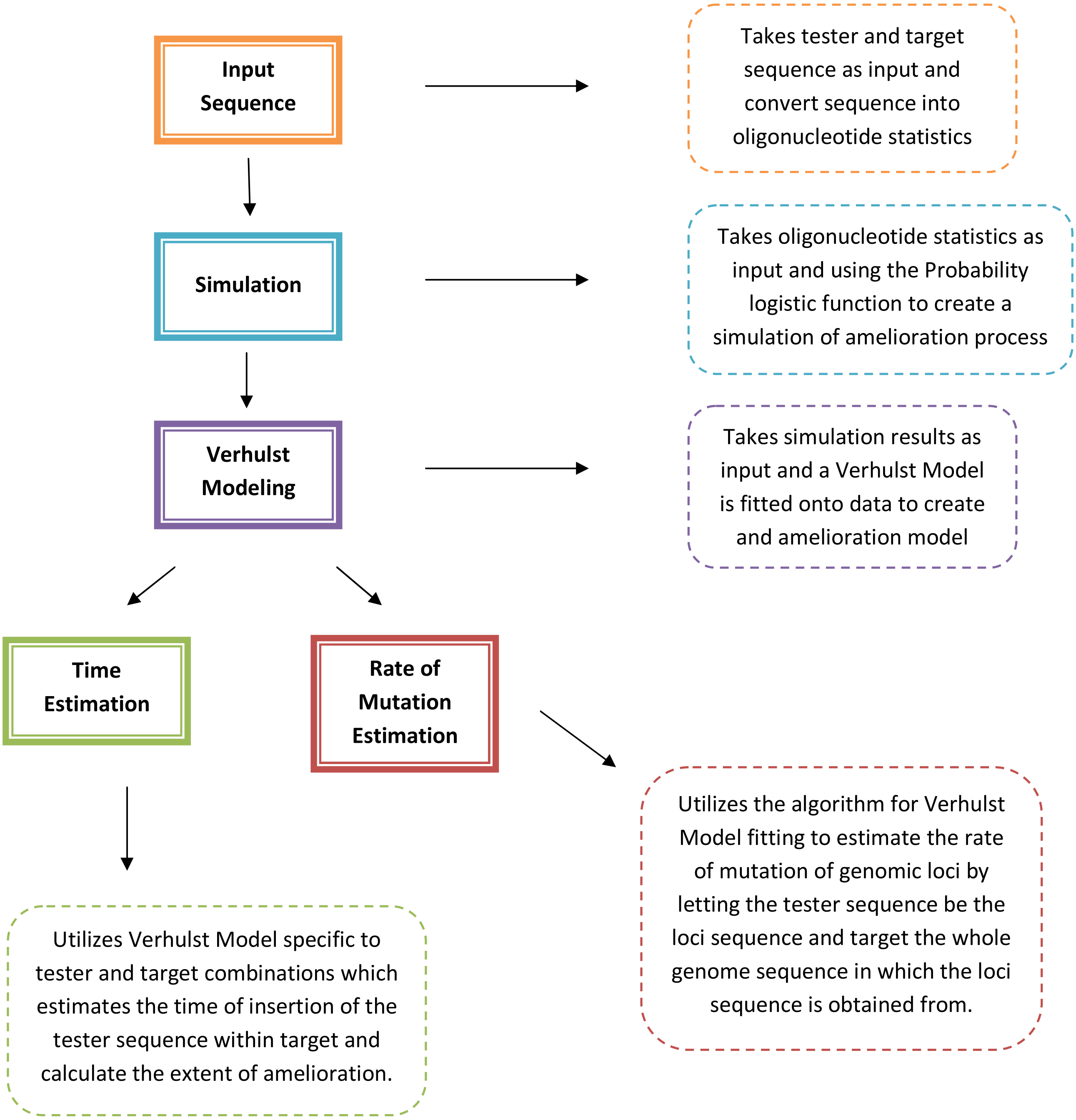
Simulation of amelioration process was done using compositional methods on each combination of GI (tester) and recipient (target) whereby k-mer words (di-mers, tri-mers, tetra-mers) were calculated and ranked based by their frequencies in descending order of oligonucleotide usage (OU) (Bezuidt et al., 2009). A logistic probability function was then used to convert the ranked frequencies into a probability which gives the likelihood that at any given position the nucleotide will be substituted into another. A program on Python to simulate the amelioration process of generations with a given mutation rate was designed and in turn simulates the amelioration process for the underlined GIs. An amelioration model was then derived and fitted to the standard Verhulst model, which in general used in population dynamics. The standard Verhulst model was fitting to amelioration process was due to its sigmoid curve shape that fit the basic assumption of directional mutation.

The parameters within the model were also well suited for the simulated data and represent a good fit for the sample simulations. The program predicts a graduate merging of the insert's OU profile with those of the host genomes that would stabilize at some level of pattern similarity. The dynamics of this process and the level of stabilization depend on the rate of mutations in the tester organism as well as the composition of the tester and target sequence. Using statistical methods, a regression model was made as a simplification in creating the amelioration model with the above three parameters which can be used for any bacterial organism.

The resulting amelioration model was also used on 4 distinct GIs from the same origin sequence (Klockgether, 2007) towards the same target. The time of insertion estimate was reasonable for all four distinct GI sequences which prove that the algorithm is suitable for estimating of the time lapsed after GI acquisition by a bacterium. The algorithm was also modified to estimate the mutation rates in different organisms and genomic loci though the results were inconclusive with further improvements needed within the method.

Methods

Flow Chart



Takes tester and target sequence as input and convert sequence into oligonucleotide statistics

Takes oligonucleotide statistics as input and using the Probability logistic function to create a simulation of amelioration process

Takes simulation results as input and a Verhulst Model is fitted onto data to create and amelioration model

Utilizes the algorithm for Verhulst Model fitting to estimate the rate of mutation of genomic loci by letting the tester sequence be the loci sequence and target the whole genome sequence in which the loci sequence is obtained from.

Utilizes Verhulst Model specific to tester and target combinations which estimates the time of insertion of the tester sequence within target and calculate the extent of amelioration.

Sequence Data and Oligonucleotide Usage Statistics

Four known genomic islands (GI), labeled under tester, were used to model their amelioration process towards compositional profiles of genomes of organisms representing distant taxa and different GC content, i.e. *Bacillus subtilis* 168 (BS), *Pseudomonas aeruginosa* PA01 (PA), *Escherichia coli* K12 (EC), *Xylella fastidiosa* 9a5c (XF) and *Streptomyces griseus* NBRC 13350 (SG). 5 Genomic islands PAGI1, PAGI2, PAGI3, PAGI4 and PKLC102 were used to estimate the time of insertion (Klockgether et. al., 2007). Detailed statistics are within table 1 below. GIs used in this study was identified and obtained from SWGIS Pre_GI database within the SeqWord Project [1]. Similarly, the five target organisms sequence data (Fasta format) were also obtain in this manner. Genome sequences chosen were of similar bp sizes such that it reflects the true amelioration process whereby an insertion of GI is within the recipient. The sequence file was then used in another program made using python script Oligonucleotide Pattern Evolution Project (OPEP).

Table 1. Detailed statistics of genome sequence used within study

Tester						**Target**		
Host	**NC**	**GI#**	**Start**	**End**	**Length**	**Start**	**End**	**Length**
***Bacillus subtilis* 168**	NC_000964	9	2146000	2258655	112655	1000009	1100009	100001
***Escherichia coli* CFT073**	NC_004431	20	3409389	3494936	85547			
***Escherichia coli* K12 substr MG1655**	NC_000913					1	100001	100001
***Streptomyces coelicolor* A3(2)**	NC_003888	33	7561923	7647787	85864			
***Streptomyces griseus* NBRC 13350**	NC_010572					1	100001	100001
***Xylella fastidiosa* 9a5c**	NC_002488					500000	600001	100001
***Pseudomonas aeruginosa* pathogenicity island PAGI 1**		1	1	51300	51300			
***Pseudomonas aeruginosa* PA01**	NC_002516					1	106370	106369
PAGI_1					51300			
PAGI_2					158230			
PAGI_3					128136			
PAGI_4					34398			
pKLC102					103609			

OPEP transforms sequence data into OU statistics which is vital in the simulation step. Combinations of GI (Tester) and recipients (Targets) are used as inputs (20 combinations of 4 tester x 5 targets) and OU statistics are calculated. These OU statistics include deviation, distance (D), variance (V) and compositional variance between tester and target (V_0) of which only deviation, V and V_0 are used throughout the algorithm (Bezuidt et. al., 2011). Deviation is a measure which calculates the logarithmic deviation of OU pattern from expected frequency of OU words. The equation is as follows:

$$\Delta_w = \Delta_{|\xi 1 \ldots \xi N|} = 6 \times \frac{\ln\left(\dfrac{C^2_{|\xi 1 \ldots \xi N||obs} \sqrt{C^2_{|\xi 1 \ldots \xi N||e} + C^2_{|\xi 1 \ldots \xi N||0}}}{C^2_{|\xi 1 \ldots \xi N||e} \sqrt{C^2_{|\xi 1 \ldots \xi N||obs} + C^2_{|\xi 1 \ldots \xi N||0}}} \right)}{\sqrt{\ln\left(\left[C^2_{|\xi 1 \ldots \xi N||0} \Big/ C^2_{|\xi 1 \ldots \xi N||e} \right] + 1 \right)}} \qquad [1]$$

Where ξ_n is any nucleotide A, T, G or C in the *N*-long word; $C_{[\xi 1 \ldots \xi N]/obs}$ is the observed count of a word $[\xi_1 \ldots \xi_N]$; $C_{[\xi 1 \ldots \xi N]/e}$ is its expected count and $C_{[\xi 1 \ldots \xi N]/0}$ is a standard count estimated from the assumption of an equal distribution of words in the sequence: ($C_{[\xi 1 \ldots \xi N]/0} = L_{seq} \times 4^{-N}$). Deviation gives a relative measure in terms of a logarithmic normal distribution of abundant or rare a specific OU word is within the sequence. A low deviation measure (negative value, less than expected) implies rare OU word within sequence while high deviation measure (positive value, higher than expected) implies abundance of OU words within pattern.

Variance (V) is calculated based on deviation in which the variation of deviation based on the whole sequence is calculated as seen in equation [2].

$$RV = \frac{\sum_{w}^{4^N} \Delta_w^2}{\left(4^N - 1\right) \times \sigma_0} \qquad [2]$$

Where σ_0 is the expected standard deviation of the word distribution in a randomly generated sequence which depends on the sequence length given by:

$$\sigma_0 = \sqrt{0.02 + \frac{4^N}{L_{seq}}} \qquad [3]$$

The compositional variance (V_0) calculates the variation of the deviation between two OU patterns tester and target. Equation [2] is used with the only difference being the deviation value used is calculated as $\mathrm{Dev}_{\mathrm{Tester}}$ - $\mathrm{Dev}_{\mathrm{Target}}$.

Figure 6 is a graphical representation of OPEP program with. Two sequences are used as input situated in the top left and middle block while the difference in their OU pattern is shown within the top right block. Each OU is also shown in a block plot, each block representing a k-mer word, where the colour shows how frequent the k-mer word is present in the sequence (red showing k-mer word present in sequence more than expected, blue being the opposite). The intensity of the colour reflects how abundant/rare the word is represented. All OU statistics for the specific sequence are shown underneath. These OU statistics are then stored as variables to be used in the simulation step where the python script is already a part of the OPEP program.

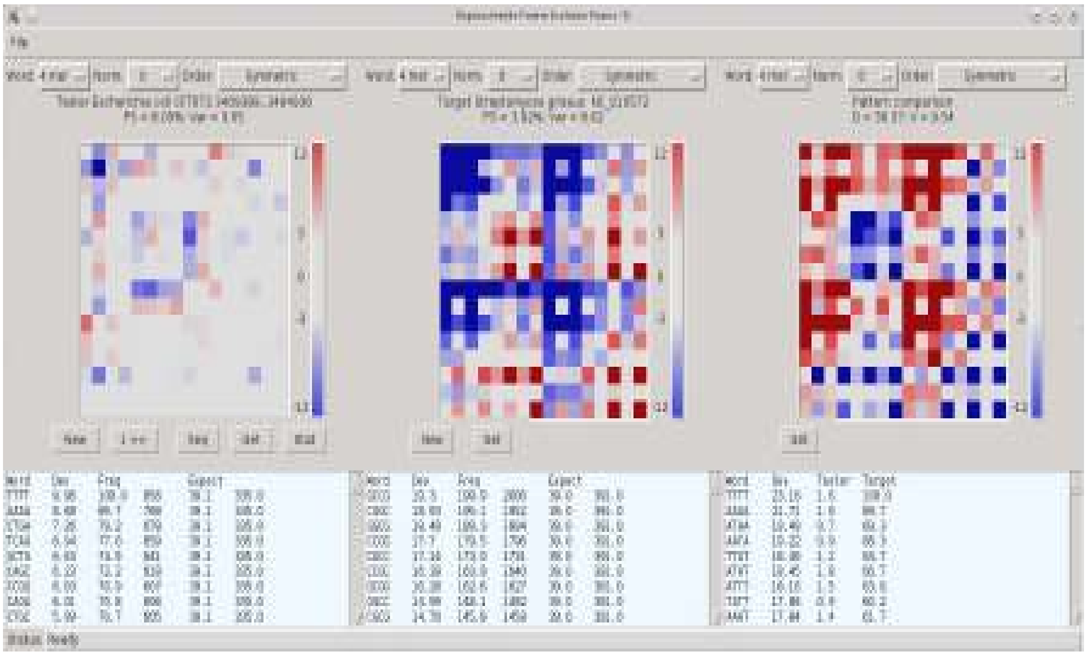


Fig. 6. Example of a screenshot of the OPEP program displaying a combination of tester (E.coli) and target (S. griseus). The left hand side represents the tester, middle the target and right the difference between two patterns. Tetra-mer word pattern is shown above where the OU statistic is shown in text below the block plot. Dev representing deviation (Equation 1) is a measure of frequency of word occurred in sequence that deviates from expected. In the right text block, the dev parameter shows the deviation measure between two patterns. The words are also in descending order whereby the highest dev is ranked first. Other parameters such as Pattern skew (PS), internal variance (Var, equation 2), variance (V) and distance (D) is also present. Distance shown on the top right corner represents the absolute distance between ranks of oligonucleotide in the two patterns. PS is a particular case of distance measure which calculates the distance between direct and inverse strands of the same DNA. Lastly var shows the variability of pattern within the sequence and V is the variability between sequences. The block plot gives an easier representation of the frequency measure of each k-mer word and is colour coded. Putting the cursor over each block also gives the status of the word in detail.

Probability Logistic Function

The simulation process uses a special derived function (Probability logistic function) which converts a deviation measure into a probability that at any given position, a specific nucleotide can be substituted by another. The deviation used here is slightly different from equation 1 where the deviation measure here needs to only consider the nucleotide instead of the OU pattern and uses the deviation value based on the difference between tester and target sequence (Dev_{Tester} - Dev_{Target}). Consider the following example sequence …TGGTGGGTCGTGTAGG… where deviation at nucleotide T needs to be calculated for the given sequence, the following OU words statistics up to tetra-mer GGGT, GGT, GT, GGTC, GTC, TC, GTCG, TCG, TCGT are calculated. A win score is calculated for each possible permutation of substitution i.e. for GGGT:

OU Word	Prior deviation	Posterior deviation	Win score (Posterior – Prior)
GGGT	-5.22 (Dev_{GGGT})	-5.22 (Dev_{GGGT})	0
GGGC	-5.22 (Dev_{GGGT})	-4.71 (Dev_{GGGC})	0.51
GGGA	-5.22 (Dev_{GGGT})	-0.29 (Dev_{GGGA})	4.93
GGGG	-5.22 (Dev_{GGGT})	-2.39 (Dev_{GGGG})	2.83

This is done for all related substitutions and the deviation value for a specific nucleotide is equal to the sum of all win scores of that nucleotide. I.e. for nucleotide A for the given sequence above:

GGGT → GGGA*p4
GGT→ GGA*p3
GT→ GA*p2
GGTC→ GGAC*p4
GTC→ GAC*p3
TC→ AC*p2
GTCG→ GACG*p4
TCG→ ACG*p3
TCGT→ ACGT*p4

p2, p3 and p4 in this case are the weighing parameter. By default, all weighing parameters are equal to one. This procedure is done for every nucleotide ACGT at each position of the sequence and the deviation value is a row vector of the form [Dev_A, Dev_C, Dev_G, Dev_T].

This deviation value (x) is then used in equation 4 to calculate the probability of substitution.

$$\text{Probability of substitution } = \frac{1}{3(1+e^{-(a+bx)})} \qquad [4]$$

Equation 4 is derived from the statistical logistic function and there are two reasons why this specific function is used. Firstly, the conversion range is for any x value (negative infinity to infinity) to a value between 0 and 1 (probability). In this case specifically, the function is tailored to be converted to a probability range of [0:0.33]. The reason for this is that at any given position, a nucleotide can only be substituted by 3 other possible nucleotides and also the probability of not being substituted at all. Consider an example that at position 10 of the sequence, the base is a nucleotide G. The deviation measure for position 10 is [-4.3745; 1.2864; 5.2346; -2.1465] (Deviation measure always sum up to 0) where each position in the list represents [A, C, G, T] respectively and mu equal to 0.1. Using function [1], the probability conversion becomes [0.002; 0.016; 0.779; 0.203]. Since all probabilities must sum to one, the state in which the position is at, the probability is adjusted. In this case, the substitution probability for G is 1 – substitution probability for [A, C, T].

The other logic behind the logistic function is the usage of parameter “a” and “b” for biological justifications. Parameter “a” is a function of μ such that at deviation = 0, probability of substitution must equal to average mutation rate μ. In biological sense, at no deviation, the substitution rate should be as expected which is equal to the value of the average mutation rate of the sequence. Equation of parameter “a” is as follows:

$$a = -\ln\left(\frac{1-3\mu}{3\mu}\right)$$

Parameter “b” represents the conservation of the sequence under study. A change in the “b” parameter will change the shape of the function which reflects how a small change in deviation will change the probability of substitution. For practical use, if a sequence under study is conserved, a large “b” value is used (b > 1). This will cause a small change in deviation to cause a large change in probability which is true for conservative sequences (Figure 7). The opposite apply for low “b” values (0 < b < 1). For this study specifically, a “b” value of 1 is used for uniformity.

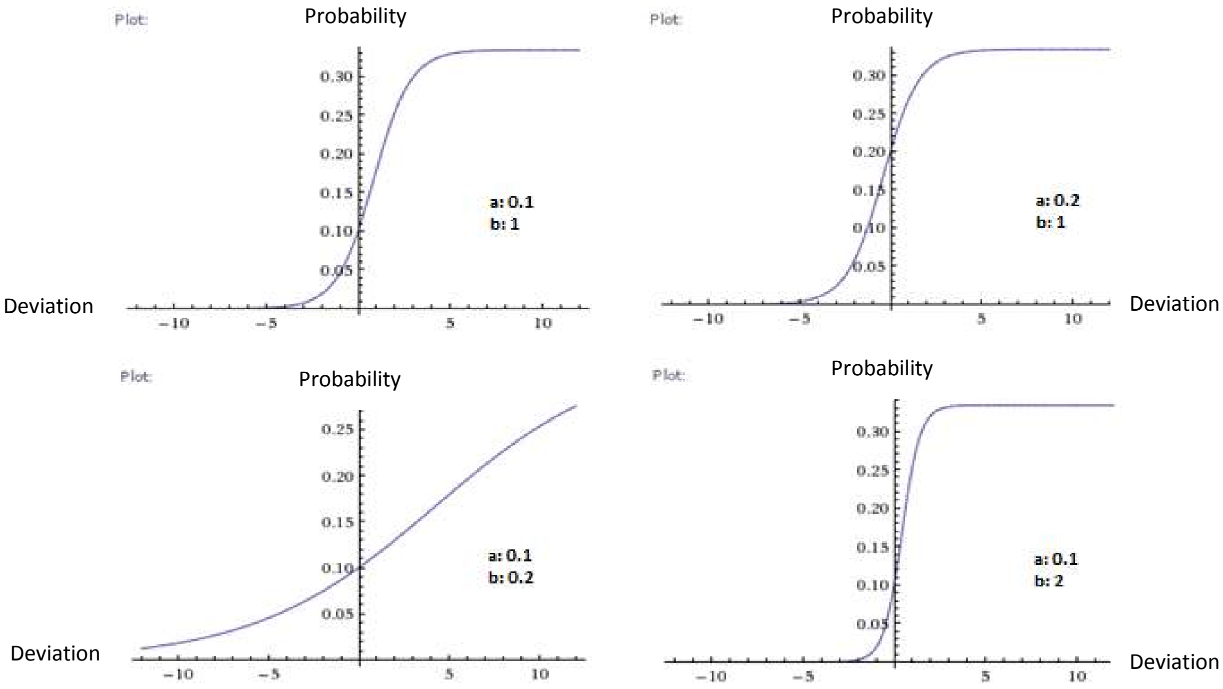


Fig. 7. Logistic probability function shifts. Changing in parameter “a” and “b” in equation [1] will change the shape of the curve to allow flexibility to reflect real life situations. Parameter “a” also known as a function of average mutation value μ. This function is especially tailored such that at x = 0 (also known as the expected value or no deviation), y intercept is equal to μ. Meaning no abnormal mutation should occur and be as expected hence at this point, the mutation rate reflects the μ value. The shift in parameter “a” will shift the y intercept up and down which is shown in the top left and right graph. In biological sense, this shift will change the subset of probabilities in which the over/under represented words can be converted into. Parameter “b” represents the conservativeness of the sequence. A large “b” value will shrink the graph and an increase in a small range of deviation will increase in a large change in probability, while the opposite occurs for a small “b” value as shown in the bottom left and right graphs. X value in this case is the deviation value for each [A, C, G, T] at a specific position.

Results

Simulation Results

In order to identify possible amelioration models, a simulation of the amelioration process is needed first to generate data for the fitting procedure. Simulation step starts with user input parameters which is vital in controlling the results you are given. The parameters include the number of iterations for the simulation process, the average mutation rate for the sequence under study and the weighing parameters of k-mer words (Figure 8). The number of iterations forms the core of the simulation process and it determines how many generations the sequence will undergo allowing random nucleotide mutations. The average mutation rate (μ) parameter directly correlates with the amount of mutation (Figure 7) and k-mer weighing parameter determines the weighting of the k-mer patterns on the deviation measure. The "save report" option allows you to print the results onto a textfile with advanced option of saving the results after every N iteration. For this study, a μ value of 0.00001, 0.00004, 0.00008 and 0.0001 are used for all combinations of tester and target and the iteration is set at 2000. Normalization of k-mer remains unchanged and the report is saved after every 100 iterations.

The simulation procedure then proceeds by calculating the necessary OU statistics first. These statistics include the deviation measure for each OU word pattern (equation 1), the variance of tester and target sequence based on the deviation measure (equation 2) and the compositional variance (V_0) between the two sequences (equation 3). At each iteration of the simulation process, the deviation value at each specific position is calculated based on equation 4 as well as the probability of substitution for each nucleotide based on the probability logistic function. A random number is rolled between zero and one and depending on the random number it will determine which nucleotide the base will substitute into. For example, the base position is G and the probability of mutation is [0.002;0.016;0.779;0.203] (based on previous example), therefore the cumulative mutation probability becomes [0.002;0.018;0.797;1] for [A,C,G,T] respectively. If the random number falls between 0 and 0.002 then a mutation of G to A occurs and similarly for other nucleotides. This process is done for every single base position within the sequence and with every new iteration, this procedure is repeated starting from the calculation of OU statistics (since OU statistics are different for the new sequence). The simulation process ends when all iterations specified by user are done.

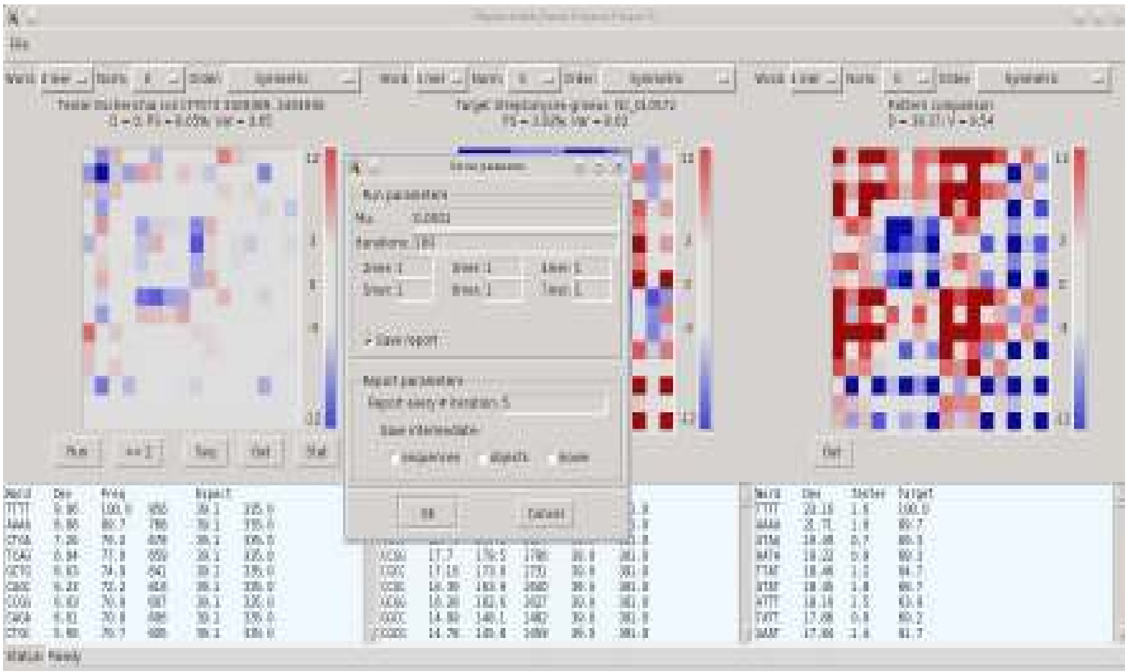


Fig. 8. There are several important parameter settings for the simulation process. µ representing the average mutation rate of the sequence which is inserted by the researcher depending on the data. For this research specifically, µ values of 0.00001, 0.00004, 0.00008 and 0.0001 are used. Iteration parameter determines how many generations you want the sequence to undergo mutation. Each iteration calculates new substitution probabilities depending on the newly calculated deviation values of the new sequence simulated. The k-mer (k representing 2-7 in this case) option allows you to specify the weighting of a specific k-mer pattern. Save report option allows you to save the final simulation report in a text file. The advanced option under save report allows you to specify the intervals in which the intervals get reported and the option of saving the sequence is also available.

Initial simulation testing was done on the combination of P. *aeruginosa* 01 GI tester against a P. *aeruginosa* genomic fragment that is different from the sequence of GI as target. Different combinations of iteration (100, 300, 500, 1000, 5000, 10000) and µ (0.000001, 0.00001, 0.0001, 0.001, 0.01) were used to test the efficiency of the simulation program. As the iteration value increases, computation time increases exponentially (100 iterations – 8 hours, 10000 iterations – 4 weeks) while µ does not make a significant difference in computation time. Another factor that influences the computational time is the internal variance of the tester sequence (E. *coli* – 6 hours for 100 iterations and S. *griseus* takes 10 hours for 100 iterations) where lower internal variance sequences tends to run quicker than higher ones. µ parameter does however impact the logical part of the simulation and two situations could occur. High values of µ tend to allow too many substitutions per iteration and the amelioration process proceeds unnaturally fast (Figure 9) hence close to random which could be seen as the program's limitation. Similarly, low values of µ will have the opposite effect. From the various simulations done with different µ values, the range of [0.00001; 0.0001] seems to work the best.

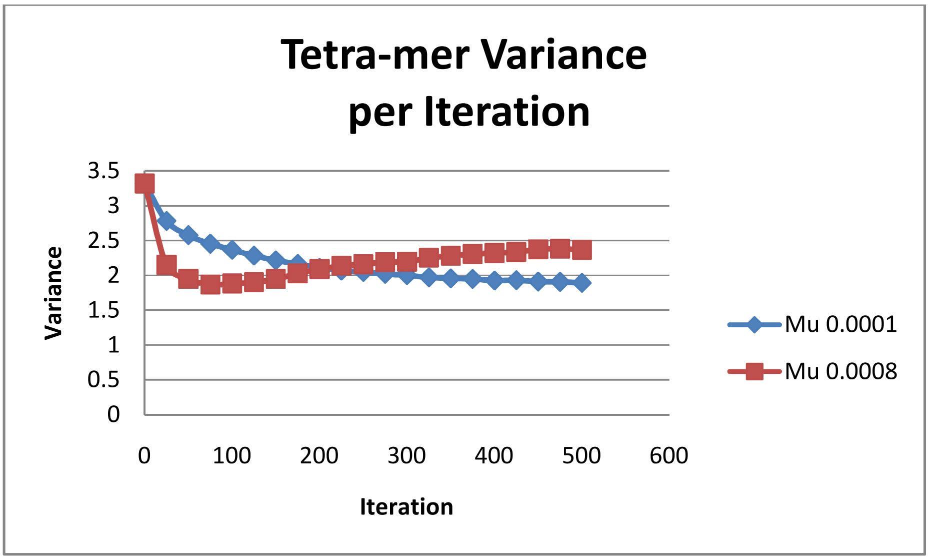


Fig. 9. Comparison between different µ values influencing the variance measure. Higher µ value will increase the number of substitution per iteration (Increasing µ increases the probability of substitution at each base position based on the logistic probability function) therefore causes the difference in steep decline between the red curve and the blue curve in the first 100 iterations. Due to the higher µ, red curve reaches the minimum variance value much quicker and therefore random substitution occurs. This is due to tester sequence reaching a similar state as the target sequence (minimum variance value) which at higher µ, unlikely substitutions will become more likely (upward slope shown from 100 iteration onwards) therefore creating a situation of mutating away from target. The blue curve is on the other hand is a better representation of the amelioration process compared to the red curve where the process is smoother and better fit to biological applications.

After the initial simulation run, four tester and five target combinations were run with different µ values for each combination. A total of eighty simulations were done and each simulation was analyzed individually. All simulation results shared similar trends of high variance decrease in the initial iteration period (usually 0 to 500 iterations) where the first 100 iterations showed the greatest different in variance. The decrease gradually diminishes and reaches a limit value which is similar to figure 11's blue curve. Each simulation result of tester and target combination is different (variance decrease rate, limit value) but a general trend can be seen where combinations with similar compositional pattern (low V_0) tend to have lower limit value and lower variance decrease rate while for the opposite case, the effect is the other way around (highly different composition between tester and target, high V_0, will tend to have a high limit value with high variance decrease rate).

We found that the general trend of the simulation data follows a logistic curve much like a Verhulst model. Possibly applicable to simulate the amelioration changes in DNA composition of horizontally acquired islands towards the host genome pattern, however we found out that high rate of substitutions (μ) may significantly alter the sequence from a Verhulst based expectation. It was necessary to evaluate the appropriateness of the Verhulst model for different mu-values and tester-target combinations by investigation residual values. These results are presented in the next section.

Verhulst Model Fitting

Verhulst model is well known for its uses within the biology field for modeling population growth (Horowitz et al., 2010; Koseki and Nonaka, 2012). The sigmoid curve which defines the model is useful in this case where an extreme increase in the beginning shows the signature of the amelioration process (directional mutation) and the gradual decrease at the end which shows the similarity in the resulting sequences. This model shows much similarity to what we see within the results from the simulation data. Using the output from simulation results, we test the possibility of a Verhulst based model of amelioration process. The simplest way to do this which is also used within this study is to derive the differential equation of the simulated data and see if the differential equation matches the Verhulst model hypothesis. The Verhulst differential equation [5] and standard equation [6] is shown below:

$$\frac{dV}{dt} = gV(1 - \frac{V}{K}) \qquad [5]$$

$$v(t) = \frac{V_0 \times Ke^{gt}}{K - V_0 + V_0e^{gt}} \qquad [6]$$

Where g controls the slope of the differential equation, K is the maximum capacity of variable V. In the case of equation [6], additional parameters are V_0 (not to be mixed with compositional variance) which is the initial value at time t=0 and t equals the time period.

Using the compositional variance between tester and target from the simulation data of di-mer, tri-mer and tetra-mer separately (highlighted yellow in Table 2) against iteration as y and x values, linear regression is done to fit an equation. This could be viewed as the composition change of the sequence over time. Suppose the iteration dataset consists of $\{t_0, t_1, \dots , t_{n-1}, t_n\}$ and similarly the V_0 dataset being $\{V_0, V_1, \dots , V_{n-1}, V_n\}$ (V could be either di-mer, tri-mer or tetra-mer). The empirical differential dataset is calculated by fitting a linear line through "n" points and taking its derivative. "n" being equal to {2,3,…,N, N+1} and N equal to the cutoff value. The cutoff value is calculated as the point in which the variance is reaching equilibrium or no longer is decreasing ($V_n - V_{n-1} > 0$ or very close to zero). The logic behind the cutoff value is that we are trying to determine the limit value which is one of the parameters of the Verhulst Equation (K) and all values beyond the cutoff value will cause the differential dataset to become skewed (derivative very close to zero due to a large number of points being relatively close to each other in value) and hence estimating K becomes increasingly more inaccurate.

Table 2. Simulation Output

Iteration	Mutation	Delta	2D	2V	3D	3V	4D	4V
0	0	0	8.82352	6.23378	12.16346	5.58532	12.38448	5.31588
100	6750	6750	6.61764	4.42955	9.75961	4.2549	9.93433	4.30297
200	8908	2158	5.88235	3.99178	8.84615	3.93263	9.50571	4.07895
300	10498	1590	5.88235	3.71179	8.36538	3.71797	9.19868	3.93205
400	11837	1339	5.88235	3.49119	8.02884	3.54465	8.99805	3.80521
500	12963	1126	4.41176	3.30584	7.74038	3.39143	8.80654	3.69319
600	13955	992	4.41176	3.16697	7.45192	3.27523	8.6363	3.60998

Iter: Iteration; Mut: Cumulative mutation frequency; Delta: Mutation occurred during the iteration period; D: Distance V: Variance

For example, let N be equal to 20 and the first row of the differential dataset will be equal to the gradient of the set $\{(V_0:t_0), (V_1:t_1)\}$ and second row will become the gradient of $\{(V_0:t_0), (V_1:t_1), (V_2:t_2)\}$. This process will go on until all points (N) are taken into calculation. This step is done separately for di-mer, tri-mer and tetra-mer dataset. Figure 10 shows a visual representation of this step.

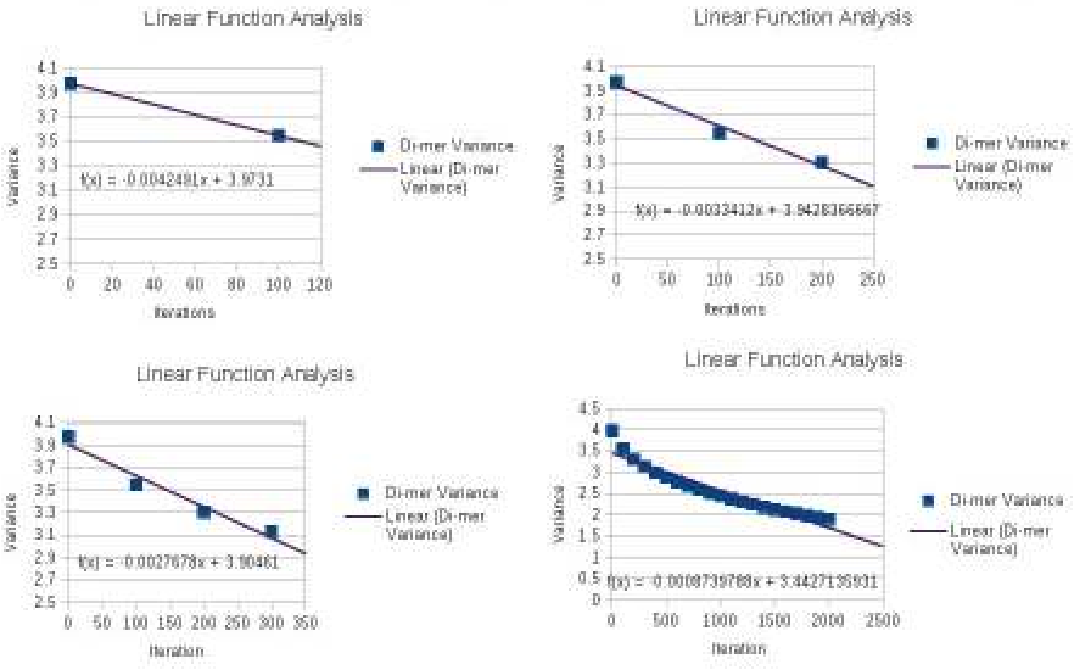


Fig. 10. The process of creating the dataset for fitting the differential equation. Using the di-mer variance data points (Variance being the variable under study) as the points for line fitting. Initially two points are used and a line is fitted (top left), and after each fitting, another point is added (top right, bottom left) until there are no more points available (bottom right). For each line fitting, the derivative is used as a data point for the differential function.

Using the empirical differential dataset, a differential equation is fitted and parameter for the model is estimated (Figure 11). The fitted model is in the same form as the Verhulst differential equation [5]. In order to derive the Standard Verhulst Equation [6], we need to derive the parameters of the equation g and K. The derived differential equation is a quadratic function where two parameters (g and K) need to be estimated. Due to its form, a linear transformation can be done which equation [5] can be re-written as:

$$y = a + bx \qquad \text{where } y = {}^{\frac{dy}{dx}}\!\big/_{V}\,, a = g, b = -\frac{g}{k}$$

Using the above simplified equation, linear regression can be performed on the empirical dataset and both parameters can be estimated (g = a, K = -a/b). A python script has been written to perform the whole process from building the dataset to estimating both parameters. The program goes one step further in estimating the best g value (in terms of least squared error) corresponding to K. Since the simulated data follows a Verhulst differential equation model, we can assume that the Standard Verhulst Equation can therefore model the amelioration process where parameter g can explain the trend (gradual/extreme) of amelioration process, K the maximum similarity the tester and target can share in terms of compositional variance, t the time period of amelioration and V the compositional variance between tester and target.

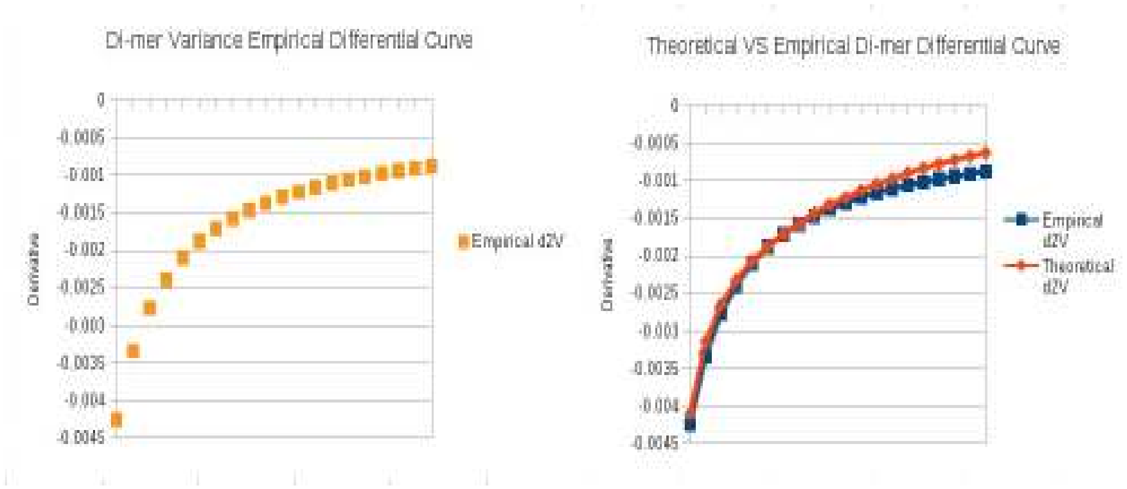


Fig. 11. Verhulst standard differential equation fitting. Using the dataset of derivatives (left), a fitting of differential equation is done (right). A python script is written for both formation of dataset as well the fitting of equation. It is shown on the right graph the difference between the empirical derivative (blue) against the estimated (red).

Verhulst Equation was then fitted to all combinations of tester, target and µ giving two more parameters g and m where m is equal to g/k. The results were recorded down into a table listing each combination with their corresponding µ values (Table 3). The resulting table was analyzed independently for each parameter (g and m) and their correspondence to the input parameters (Tester, Target, µ). Residue values were also recorded for each combination as absolute value difference between estimated equations against simulation data. The residue dataset will give more insight into how well the Verhulst Equation actually fit to the simulation of amelioration process as well as what factors might influence the fitting of the equation.

Table 3. Verhulst Model Fitting Parameters

		Dimer		Trimer		Tetramer	
Tester/Target	**µ**	**m**	**g**	**m**	**g**	**m**	**g**
PAGI	0.00001	0.001027	0.002034	0.001107	0.002628	0.000918	0.002566
-	0.00004	0.001765	0.002859	0.001976	0.004129	0.001310	0.003145
Ecoli	0.00008	0.001681	0.002236	0.002202	0.004293	0.001917	0.004506
	0.0001	0.001603	0.001915	0.003472	0.007150	0.002957	0.007223
PAGI	0.00001	0.000160	0.000148	0.000413	0.000825	0.000551	0.001258
-	0.00004	0.000424	0.000314	0.000927	0.001558	0.001297	0.002711
9a5c	0.00008	0.001004	0.001014	0.001600	0.002720	0.002633	0.005502
	0.0001	0.001122	0.001133	0.001997	0.003489	0.002608	0.005492
PAGI	0.00001	0.001264	0.003804	0.001379	0.004922	0.001156	0.004392
-	0.00004	0.001013	0.002147	0.002205	0.007278	0.001911	0.006652
Subtilis	0.00008	0.001049	0.001573	0.002548	0.007849	0.002322	0.007661
	0.0001	0.001010	0.001271	0.002816	0.008538	0.002646	0.008640
PAGI	0.00001	0.000347	0.000346	0.000516	0.000795	0.000714	0.001428
-	0.00004	0.000727	0.000371	0.000969	0.001095	0.001282	0.002269
Aeruginosa	0.00008	0.001037	0.000539	0.001289	0.001353	0.001680	0.002889
	0.0001	0.001193	0.000585	0.001365	0.001436	0.002120	0.003604
PAGI	0.00001	0.000696	0.001446	0.000734	0.001717	0.000807	0.002385
-	0.00004	0.000948	0.001299	0.000919	0.001488	0.000916	0.002024
Griseus	0.00008	0.001449	0.001710	0.001390	0.002126	0.001297	0.002801
	0.0001	0.001599	0.001837	0.001674	0.002506	0.001458	0.003053
-	0.00004	0.001216	0.002758	0.001109	0.002897	0.001134	0.003408
Griseus	0.00008	0.001643	0.003024	0.001921	0.004434	0.002043	0.005856
	0.0001	0.001756	0.003011	0.002188	0.004981	0.002365	0.006774

PAGI: P. *aeruginosa* Genomic Island, Ecoli: E.*coli* K12, 9a5c: X. *Fastidiosa* 9a5c strain, Subtilis: B.*subtilis* sub 168, Aeruginosa: P. *aeruginosa*, Griseus: S.*griseus*, m and g are both parameters of the Verhulst Equation where m = g/k (see methods). B. *subtilis* Genomic Island, E. *coli* K12 genomic island and S. *coelicolor* genomic island in combination with five targets are not displayed in this table (See Appendix Table 1).

Box and whisker plots were made with 240 residue values made from 4 tester, 5 target and 4 µ combinations. The first plot is divided into different mu and K-mer combinations and their influence on the equation fitting step (Figure 12a). From the plot, a general trend of increasing in µ value decreases the residue of the fitting. This is seen for all K-mer sizes with respect to the mean residue value (represented by the black line inside the blue box). This could be explained by the low substitution rate caused by the low µ value which in turn causes a higher estimation of capacity value K based on the simulation data. With a higher K value, g parameter which depends on K will become lower (g parameter determines the slope/steepness of the graph hence with a higher K value; the graph will become less steep to compensate).

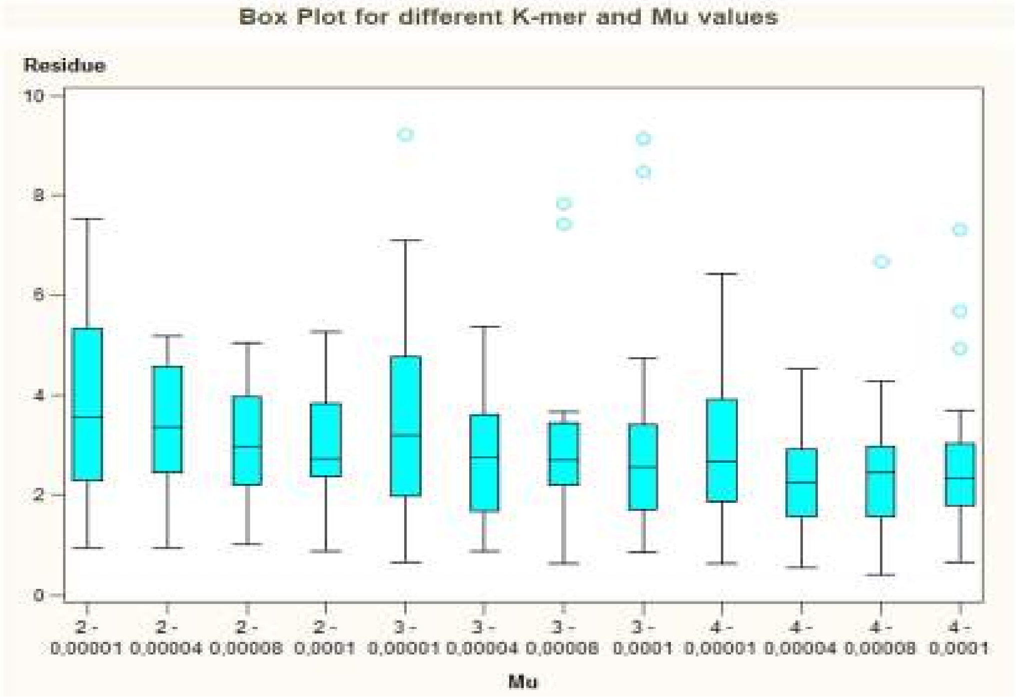


Fig. 12a. Box and whisker plot of residue values corresponding to k-mer and µ combinations. 2, 3, 4 in the graph shows di-mer, tri-mer and tetra-mer respectively while the decimal values represent µ. It is shown in the graph above that lower µ values fit poorly with the Verhulst Equation and gradual increase in µ decreases the residue value (calculated as simulation variance – estimated variance by Verhulst Equation). Some outliers occur at higher µ values which correspond to poorly matched tester and target combinations (See figure 12b). There are more outlier residue values for tri-mer and tetra-mer due to a higher capacity value K value compared to the di-mer (See Figure 13). Therefore the capacity value K is achieved quicker in the tri-mer and tetra-mer case which allows more random substitution than di-mer hence a higher residue value.

Another factor visible within figure 12a is that higher µ values caused more outliers within tri-mer and tetra-mer residue values. Due to a small number of outliers for each set of data (maximum 3 outliers out of 20 data points), a likely assumption is that some tester and target combinations' composition is very different from each other. When this is the case, the tester will have a hard time incorporating within the target sequence and hence causes a higher K value (capacity in which the tester and target shows similarity in composition pattern). This will in turn cause a similar situation as case with di-mer data set, where a higher K value would cause a bad fitting of the Verhulst Equation and therefore the outlying residue value.

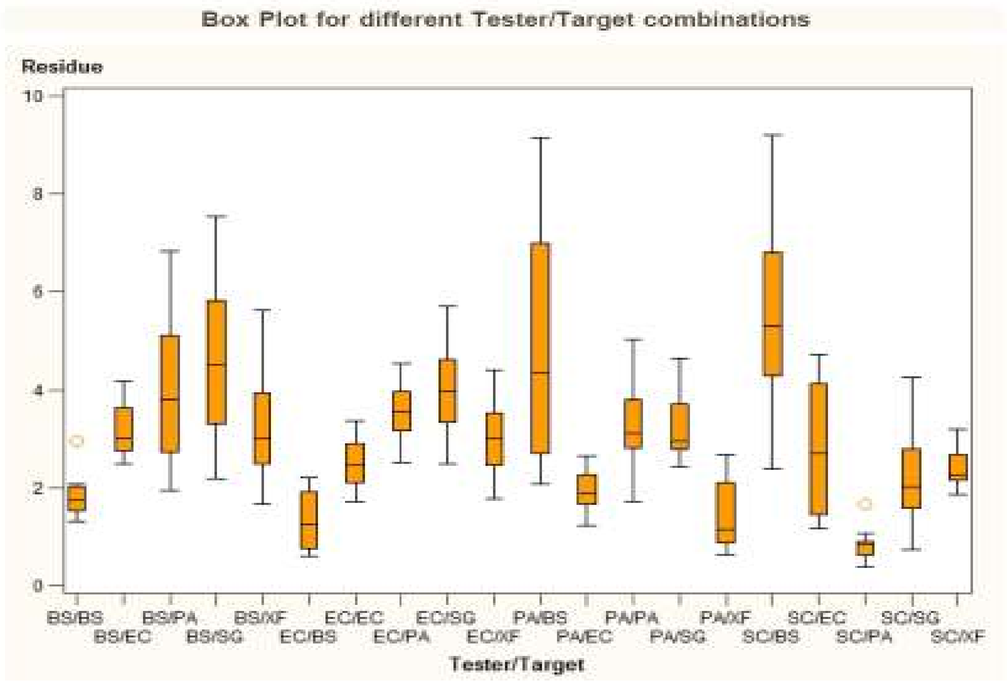


Fig. 12b. Box and whisker plot of residue between the estimated Verhulst Equation and simulation data. Each box represents a tester and target combination and the abbreviations represent: BS – B.*subtilis*, EC – E.*coli*, XF – X.*Fastidiosa*, PA – P.*aeruginosa*, SC – S.*coelicolor,* SG – S.*griseus*. The two highest residue values were 9.136 (PA/BS combination) and 9.2025 (SC/BS combination) while the three lowest were 0.4003 (SC/PA combination), 0.6146 (EC/BS combination) and 0.6636 (PA/XF combination). SC/PA combination had the smallest range of residue values as well as the smallest mean. The orange circles represent the outliers of the box plots.

The case above is supported by figure 12b where some combinations of tester and target show significantly larger residue than the others. Combinations PA/BS and SC/BS showed the highest residue while BS/SG and BS/PA were the next two highest values. Looking at these four combinations, PA/BS, SC/BS, BS/SG and BS/PA where the first represents the tester and the latter target, the organisms involved in the combination are identical in the sense that the tester and target were only swapped around. This implies that for specific tester and target combinations (PA and BS; SC or SG and BS where SC and SG are closely related) Verhulst Equation fitting is poor and hence the large residue value. Analyzing from a different perspective, combinations PA and SC against BS showed a significant difference from the rest. PA and SC share a similar internal variance (compositional parameter) while BS is significantly different from the two. Looking at the reverse situation, the lowest residue values were of combinations SC/PA, PA/XF and EC/BS. All combination share similarities in small pattern variance but PA/XF do not have similar internal variance. Therefore a hypothesis can be made that tester and target composition statistic (Internal variance, variance) and μ could be a determining factor on the Verhulst Equation parameters.

To understand which factors influences the parameter of the Verhulst Equation, Figure 13 and 14 plots the relationship between the two parameter from the Verhulst fitting and the different combinations of tester and target. There are six output parameters from the python script consisting of di-mer k (2K) and g (2g), tri-mer k (3K) and g (3g) and tetra-mer k (4K) and g (4g). Three additional parameters 2m, 3m and 4m were calculated as stated in the methods section. Figure 14 first shows the differences between the capacity values of different K-mers as well as a comparison between different combinations. It is clear that from the graph, tetra-mer K value for every combination is larger than tri-mer K value and in turn larger than di-mer. Biologically this is true where tetra-mer patterns which are much more complex than both tri-mer and di-mer patterns should have a higher variance difference (Tetra-mer pattern has 256 different combinations of 4 nucleotide length words while tri-mers has 81 and di-mers have 16).

Another interesting point in figure 14 is that combinations SC/BS and PA/BS showed the highest K values which corresponds to figure 12b. Combinations EC/SG and SC/EC also displays a high K value and since EC and SC or SG are highly different in composition, this supports the hypothesis of compositional statistics influencing the amelioration model parameters. A hidden point however, that is the two lowest K value combinations PA/PA and EC/EC shared a common factor of both testers and targets are of a similar organism. This leads to another potential factor of related species having an effect on the parameters of the Verhulst Equation.

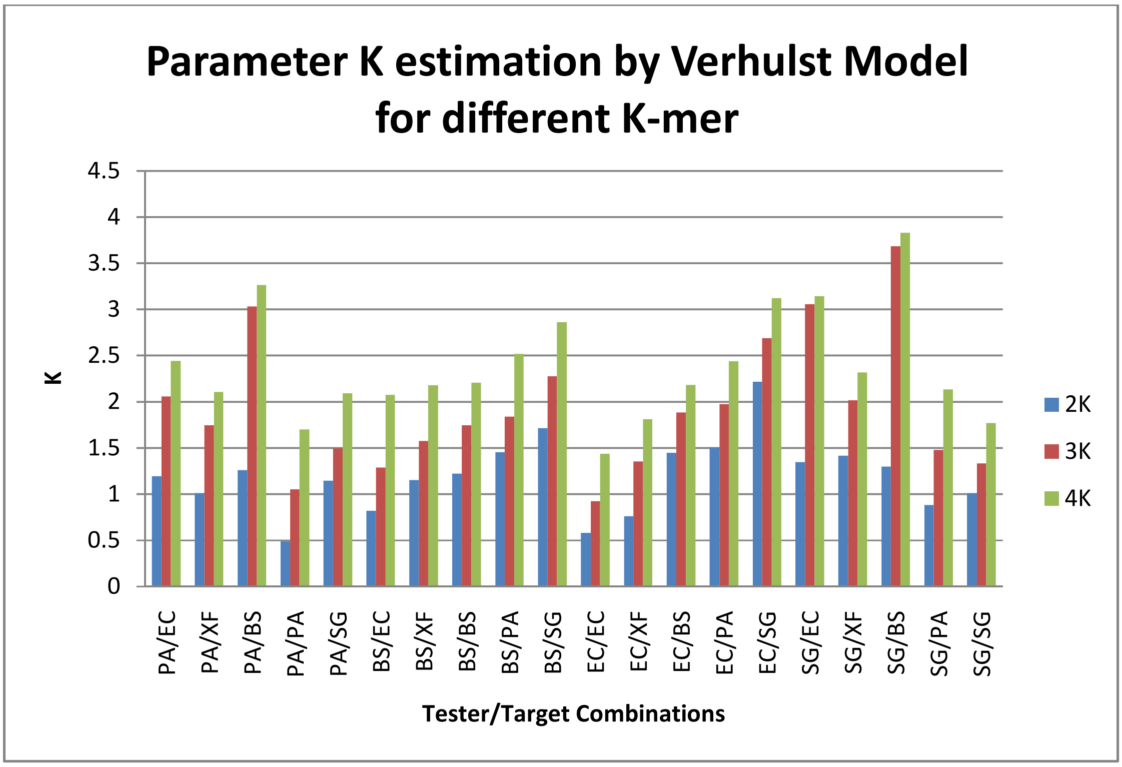


Fig. 13. A combined graph showing the different tester target combination and their estimated K parameter based on the Verhulst Equation. On the x-axis shows the different tester/target combinations and on the y-axis the estimated K value. All estimations follow the same trend of tetra-mer (green) having the highest K value estimate and while di-mer (blue) has the lowest. This is due to tetra-mer pattern being much more complex than di-mer patter (256 combinations compared to 16 respectively) and hence the pattern between tester and target variance will be further apart (represented by K). P.*aeruginosa* (PA), PA combination as tester target and E.*coli* (EC), EC combination also achieved lowest K value estimate. The common factor between the two combinations is that both tester and target are of the same organism. Though this is not definite because B.*subtilis* (BS), BS and S.*coelicolor* (SC), S.*griseus* (SG) combinations do not follow this trend. The highest K values are achieved by BS (target) combinations with SC and PA (tester). A likely explanation is that the internal variance of PA and SC are highly different. This can be further seen from the combination EC and SC therefore we can conclude that the internal variance of tester and target influences the estimation of parameter K.

Aside from parameter K, g parameter represents the rate or slope in which the Verhulst Equation takes form. A high g parameter implies a steep slope while a low value shows a shallow curve. Hence parameter g should therefore be directly linked to µ, a set constant in which determines the amount of substitution (see methods). Based on the hypothesis, parameters of the Verhulst Equation should be affected by 3 independent parameters, namely tester internal variance, target internal variance and mu. If tester internal variance does not affect the parameters, then for the same target and different tester combinations, the g and k values should be identical. Similarly for the case with the target internal variance but this is not true as shown in figure 14 where each tester and target combination showed different linear lines (x-axis µ and y-axis g parameter). This graph shows that parameter 3g is a function of tester, target internal variance and µ. This is also done similarly to the other 5 parameters 2g, 4g, 2m, 3m, and 4m (See Appendix Figure 2, 3, 4, 5 and 6).

The python script which fits the equation to the data will always use the parameters that create the least residue. This will in turn sometimes cause extreme parameter values as seen in figure 14. From figure 13, SC/EC and SC/BS had a high K value and to compensate for this, the g values were significantly different from the other combinations. This causes potential outliers which could affect the estimation of the parameter function at a later stage.

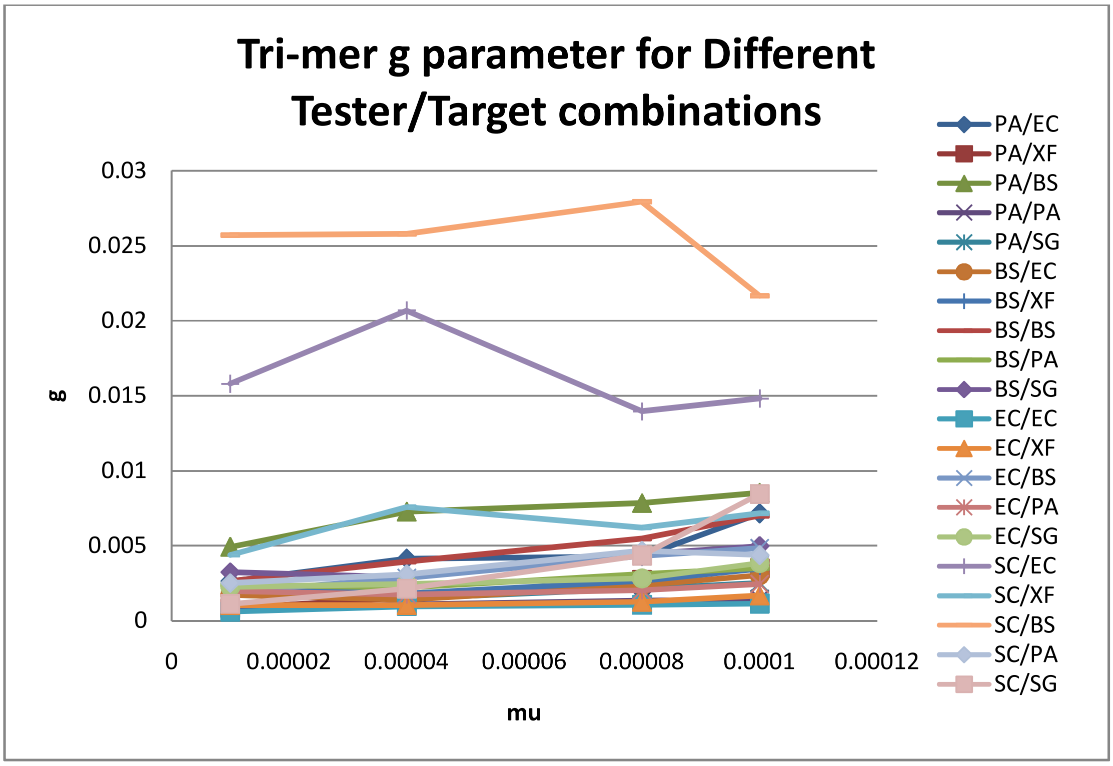


Fig. 14. Graph plot of tri-mer g parameter estimate of the Verhulst Equation of all 20 combinations of tester and target. From the above graph, tester S. *coelicolor* (SC), target E.*coli* (EC) and B.*subtilis* combination clearly is the outlier of the dataset. Due to the nature of the model fitting of the Verhulst Equation, best estimate of K and g are done on the simulation dataset. Since the pattern composition (caused by internal variance) of SC/EC and SC/BS combinations are highly different, extreme g and k values are estimated (also seen in Figure 13). Therefore these values are eliminated for a better linear fitting for the rest of the data.

It was found that the Verhulst model in general fits to simulating of the DNA amelioration process as the cumulative error residues were within the range of 0.4003 to 9.2025 of calculated absolute values. As it was expected (see the previous section), mismatches were greater when higher mu-values were set. However, contrary to our expectation, significant alterations were observed in several tester/target pairs (see Fig. 12b), particularly BS/PA, PA/BS, BS/SG and SG/BS. It indicates that the current model does not account yet for all factors influencing the amelioration process. As the model worked appropriately for the majority of tester/target combinations, we decided to use it as a working model for further analysis remembering that some improving to the model should be done in future. In the next section we consider approaches of estimating of Verhulst model parameters.

Parameter Function Estimation

From the figures of the previous section (12a, 12b, 13 and 14) we can assume that the Verhulst Model parameters for each combination share a dependency towards the characteristics of the tester and target of that combination. We tested this using all the Verhulst equation parameters from all combinations of tester and target grouped together. The initial dataset of tester and target combinations were chosen such that the different combinations were a small sample to represent the vast bacteria kingdom (each target were very different in terms of characteristic to the others, see discussion section on Parameter Function Estimation). Hence we assume and test if the general trend (dependency of Verhulst equation parameter on tester and target characteristics) will work for all bacteria combinations.

The core of the Verhulst Equation is based on its two parameters (K and g) where the others are all user inputs (V_0 and V(t)). The dataset with the parameters g and K from the Verhulst Equation were further reduced down into parameters g and m (where k = g/m therefore m = g/k). Since in the standard Verhulst Equation, K is dependent on m, hence it is more sensible to fit an equation to m rather than K. This also eliminates the dependence upon g by K hence reduces any unnecessary error within the regression process. Based on the model fitting of the 80 combinations of different tester and targets, a general trend (Figure 15) can be seen from the dataset of parameters g and m where K = g/m. To verify the truth of this relationship, a statistical multivariate linear regression test is done in SAS on parameters g and m. The variables used to test for the linear relationship are tester (V_Tester) and target internal variance (V_Target), compositional variance between tester and target patterns (V_0), µ (Mu), absolute difference between tester target internal variance (VT – VTe) and if the tester and target are of similar organism (variable equal 1 if yes, 0 if no). Different models were tested using SAS enterprise 4.3 on parameter g and m as well as different k-mers (di-, tri-, tetra-). The model with the best R-squared value was chosen using the stepwise selection method.

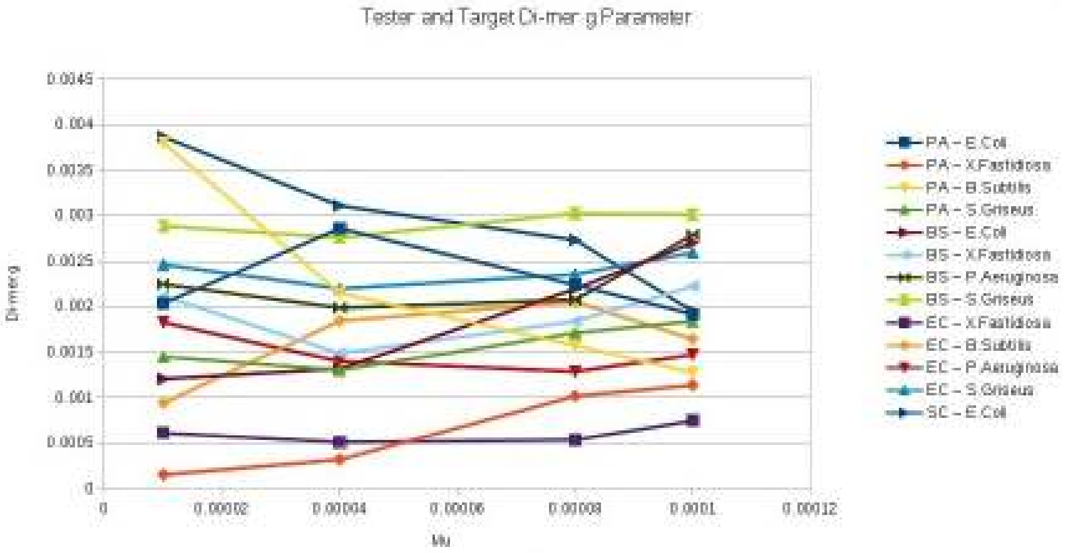


Fig. 15. Graph showing the relationship between di-mer g parameter, µ, variance of target and variance of tester. Each line showing the linear relationship of g corresponds to the other three parameter present (µ, V_{target}, V_{tester}) within the model. We can see a general trend which majority of the combinations of tester target follows (Linear Relationship).

Figure 16 shows the multivariate linear regression analysis result of parameter 4g (tetra-mer g parameter) with 80 data points each representing one combination of tester and target in association with a µ value. The adjusted R-squared value is 67.91% which shows that the variables used to model parameter 4g is not good enough as the model only explains 68% of the dataset. The bottom table of figure 15 also gives the significance of each variable within the model in which a high p value (Pr > |t|) indicate the variable being insignificant. Using a 5% level of significance cutoff, three variables namely VT – VTe, Related and Mu shown to be less useful in the model compared to the rest of the variables (V_Tester, V_Target and V_0).

Considering figure 14 where extreme outliers were present, a new dataset was constructed by filtering out these combinations. To speed up the process, selection methods (Forward, Backward and Stepwise) were used to eliminate insignificant variables which affects the adjusted R-squared value (adjusted R-squared value is calculated according to the amount of variables within the model) therefore reducing potential variance increase caused by an increase of variables within model (Figure 17). Through selection, the g parameter function chosen at 99% confidence level for each variable is in the form:

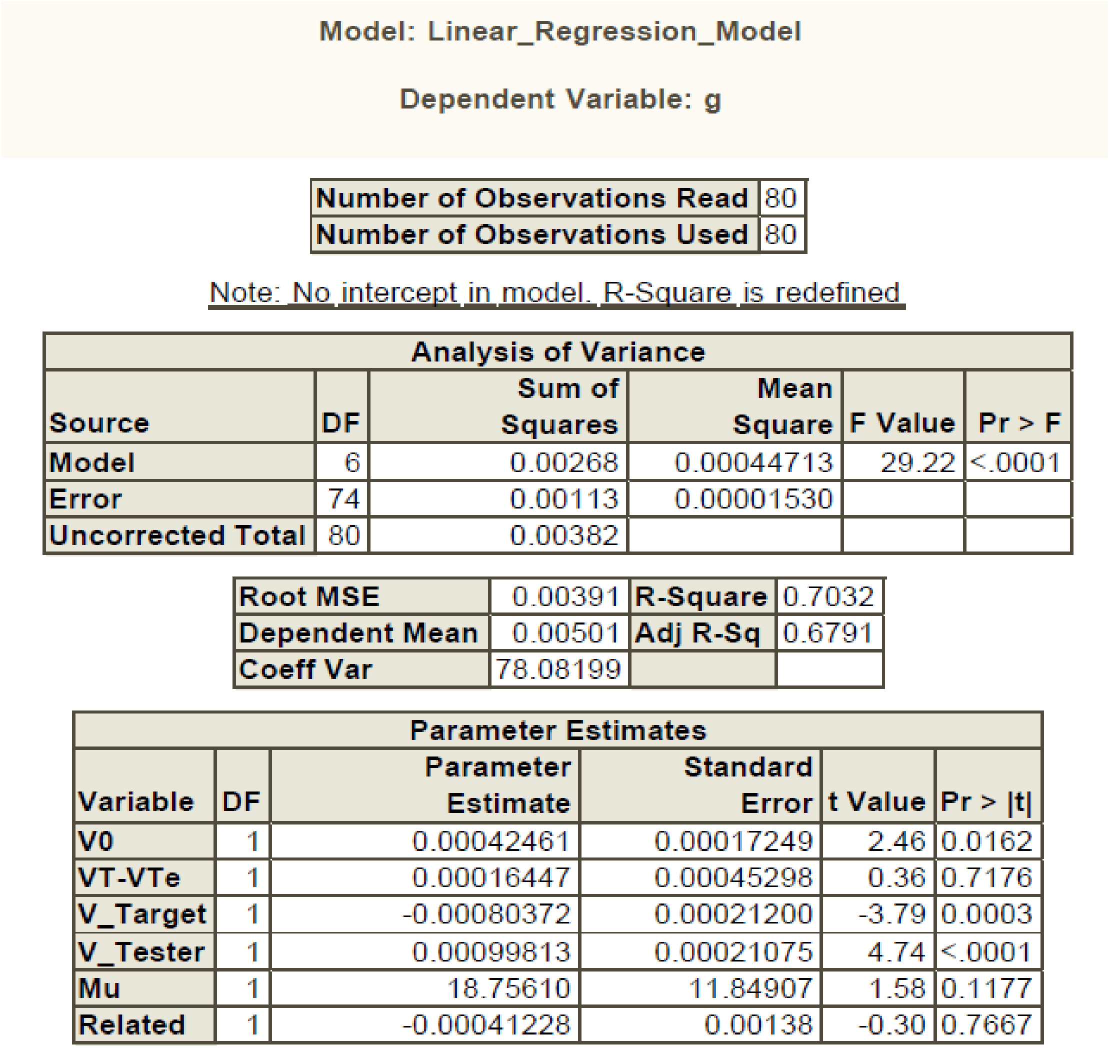
$$g = \mathrm{a} \times \mathrm{V}_{\mathrm{Tester}} + \mathrm{b} \times \mathrm{V}_{\mathrm{Target}} + \mathrm{c}\mu \qquad [7]$$

And similarly, m parameter function is of the form:

$$m = \mathrm{a} \times \mathrm{V}_{\mathrm{Tester}} + \mathrm{b} \times \mathrm{V}_0 + \mathrm{c}\mu \qquad [8]$$

Where a, b, c are all estimated constants using multiple linear regression and $\mathrm{V}_{\mathrm{Tester}}$ being the internal variance of tester sequence, $\mathrm{V}_{\mathrm{Target}}$ being the internal variance of target sequence, V_0 the variance between the tester and target sequence and μ the average mutation rate of tester.

The new fitted model achieved an adjusted R-squared value of 91.01% which is a significant increase from the previous model and is an acceptable value to say that the fitted model explains parameter g well. Other parameters (2g, 3g, 2m, 3m and 4m) were modeled the same way using multivariate linear regression (Table 4). The general form of the two parameters estimation functions are:



Model: Linear_Regression_Model

Dependent Variable: g

Number of Observations Read	80
Number of Observations Used	80

Note: No intercept in model. R-Square is redefined.

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	0.00268	0.00044713	29.22	<.0001
Error	74	0.00113	0.00001530		
Uncorrected Total	80	0.00382			

Root MSE	0.00391	R-Square	0.7032
Dependent Mean	0.00501	Adj R-Sq	0.6791
Coeff Var	78.08199		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
V0	1	0.00042461	0.00017249	2.46	0.0162
VT-VTe	1	0.00016447	0.00045298	0.36	0.7176
V_Target	1	-0.00080372	0.00021200	-3.79	0.0003
V_Tester	1	0.00099813	0.00021075	4.74	<.0001
Mu	1	18.75610	11.84907	1.58	0.1177
Related	1	-0.00041228	0.00138	-0.30	0.7667

Fig. 16.Multi-variate regression analysis with all combinations of tester, target and µ. The adjusted R-squared value (a coefficient of determination of how well the model explains the data) which takes into consideration the number of variables used is a better measure to use than the normal R-squared value. The adjusted R-squared value for the model with regards to this dataset is poor in which only 67,91% of the data is explained by the model fitted.

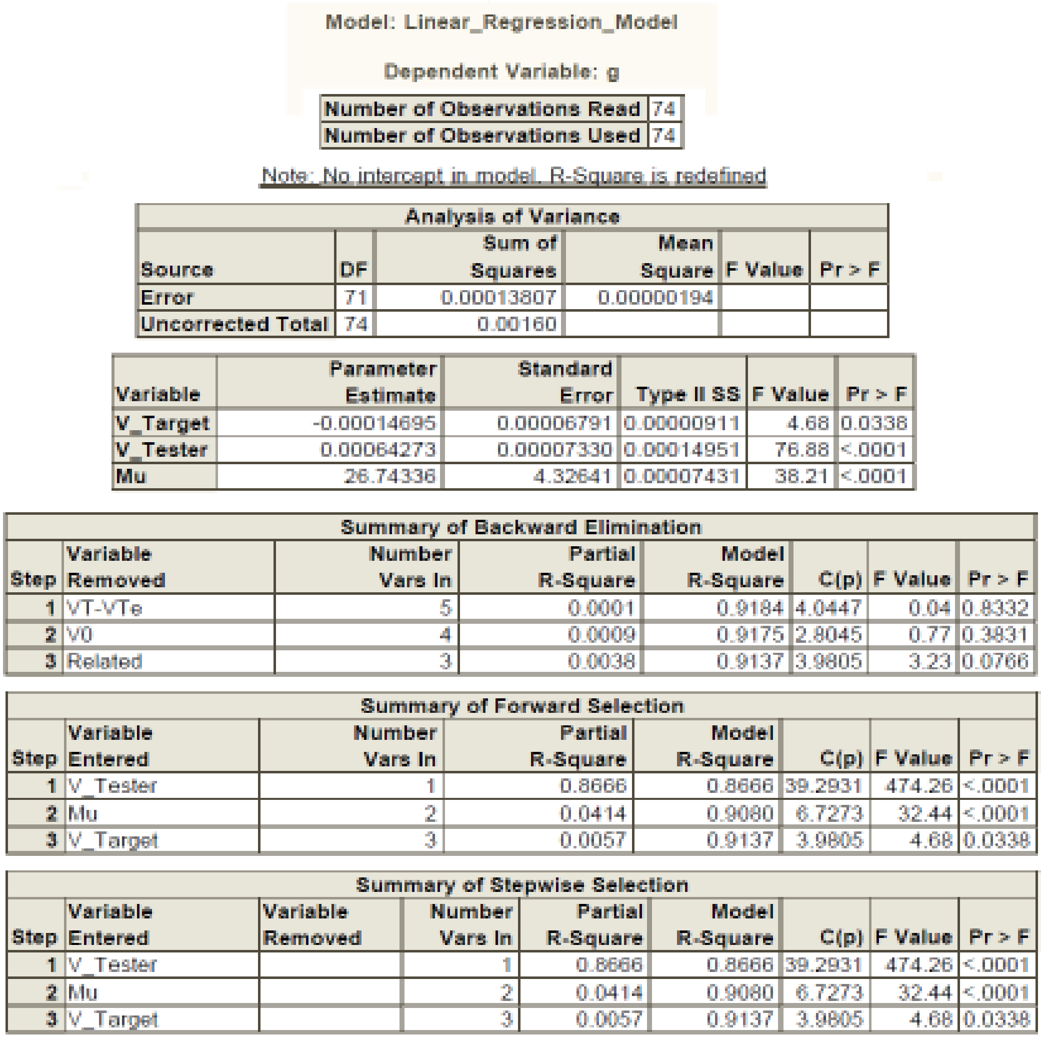
Table 4. Multivariate Linear Regression analysis results for different parameters

Parameters	Variables	Adjusted R-squared
2g	V_Tester, Mu	76.26%
3g	V_Tester, Mu	79.91%
4g	V_Tester, V_Target, Mu	91.01%
2m	V_Tester, V0, Mu	81.40%
3m	V_Tester, V0, Mu	89.40%
4m	V_Tester, V0, Mu	93.66%

2g: Di-mer g parameter, 3g: Tri-mer g parameter, 4g: Tetra-mer g parameter, 2m: Di-mer m parameter, 3m: Tri-mer m parameter, 4m: Tetra-mer m parameter, V_Tester: Tester internal variance, V_Target: Target internal variance, Mu: μ, V0: Compositional variance between tester and target pattern. SAS output in Appendix Figure 7, 8, 9, 10 and 11.

Through the different selection methods, we can see that all three of them achieved the same model (Figure 17). Since the model is identical in all three cases, we can assume this is the best model in terms of the dataset used. Three variables were eliminated (VT-VTe, V_0 and Related) in which they all only contribute less than 0.5% increase in R-squared value therefore insignificant within the model. Other three variables make up the most simplistic model in which parameter 4g can be modeled with the highest accuracy.

Taking a more in depth analysis, what if more combinations of tester and target are added, will the model be sufficient enough to explain the parameters or will more variables be added in order to compensate? Figure 18 displays three different dataset analysis each using different numbers of tester and target combinations. The three dataset consists of two tester five target combinations, three tester five target combinations and four tester five target combinations. The same multivariate linear regression analysis was done on all three dataset displaying similar results. The first dataset two tester five target combinations show a 90.39% adjusted R-squared value with variables V0, V_Tester, V_Target and Mu being significant. Similarly, second dataset three tester five target combinations model achieved 90.93% adjusted R-squared with the same variables. The only difference between the three models is the third dataset four tester five target combinations model had one less variable V0 for the model but still achieved an adjusted R-squared value of 91.01%. From these results, we can assume that the model itself is sufficient in explaining the estimated parameter with any number of tester and target combinations with the exception of the extreme outliers which the parameter function will estimate poorly.



Model: Linear_Regression_Model

Dependent Variable: g

Number of Observations Read	74
Number of Observations Used	74

Note: No intercept in model. R-Square is redefined.

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Error	71	0.00013807	0.00000194		
Uncorrected Total	74	0.00160			

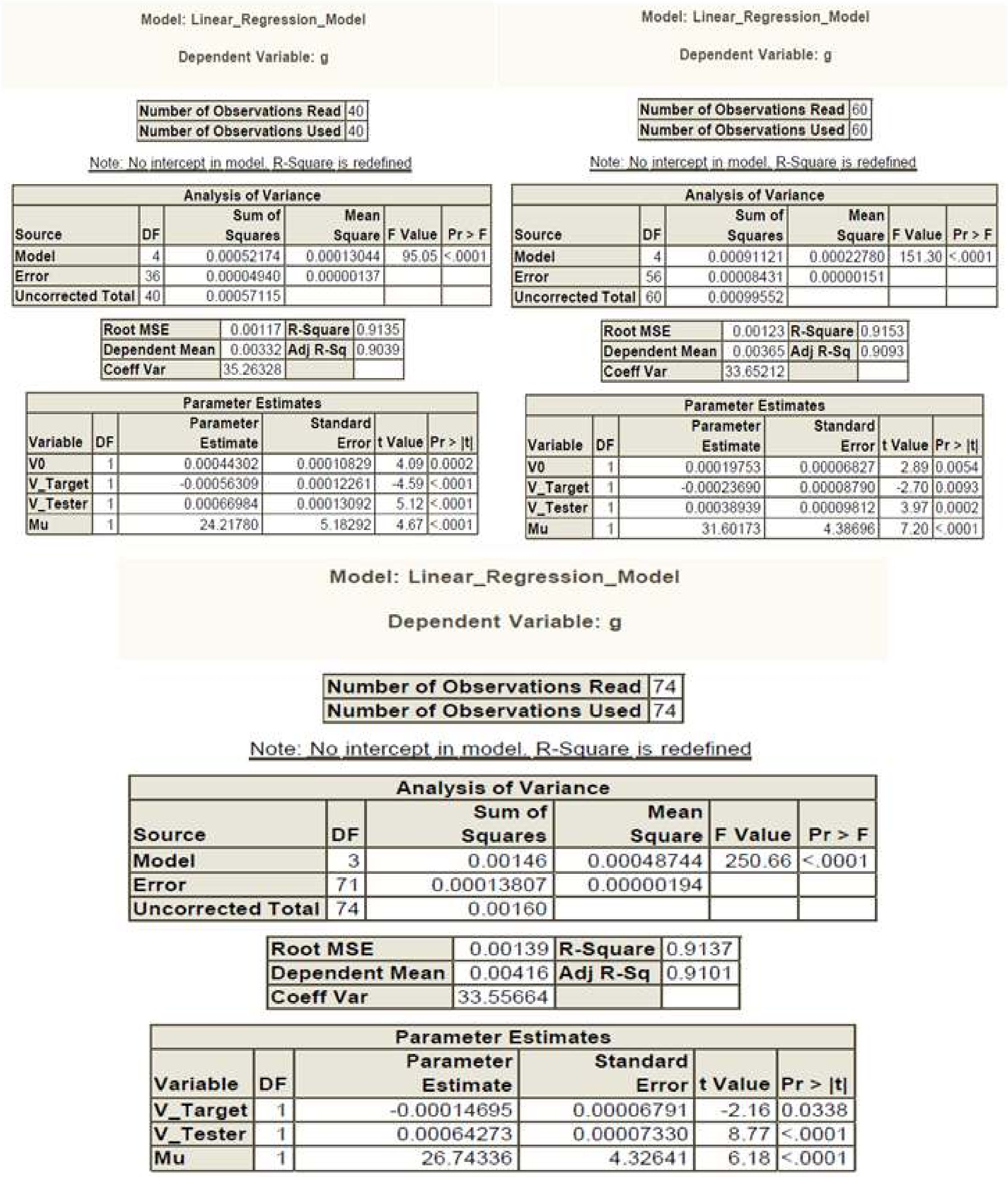
Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
V_Target	-0.00014695	0.00006791	0.00000911	4.68	0.0338
V_Tester	0.00064273	0.00007330	0.00014951	76.88	<.0001
Mu	26.74336	4.32641	0.00007431	38.21	<.0001

Summary of Backward Elimination							
Step	Variable Removed	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	VT-VTe	5	0.0001	0.9184	4.0447	0.04	0.8332
2	V0	4	0.0009	0.9175	2.8045	0.77	0.3831
3	Related	3	0.0038	0.9137	3.9805	3.23	0.0766

Summary of Forward Selection							
Step	Variable Entered	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	V_Tester	1	0.8666	0.8666	39.2931	474.26	<.0001
2	Mu	2	0.0414	0.9080	6.7273	32.44	<.0001
3	V_Target	3	0.0057	0.9137	3.9805	4.68	0.0338

Summary of Stepwise Selection								
Step	Variable Entered	Variable Removed	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	V_Tester		1	0.8666	0.8666	39.2931	474.26	<.0001
2	Mu		2	0.0414	0.9080	6.7273	32.44	<.0001
3	V_Target		3	0.0057	0.9137	3.9805	4.68	0.0338

Fig. 17. Different variable selection methods based on a 5% significance level. Six variables was taken into consideration from the hypothesis namely tester internal variance (V_Tester), target internal variance (V_Target), µ (Mu), related (if the tester and target organism is closely related, the variable takes the number one, if not then zero), V0 (Variance between tester and target composition pattern) and VT-VTe (Internal variance of target minus the tester internal variance). The final best model fitted by the three selection methods (backward elimination, forward selection and stepwise selection) are identical and shown by the first table with three variables (VT-VTe, V0 and Related) not being a significant factor explaining parameter 4g.



Model: Linear_Regression_Model

Dependent Variable: g

Number of Observations Read	40
Number of Observations Used	40

Note: No intercept in model. R-Square is redefined

Analysis of Variance

		Sum of	Mean		
Source	DF	Squares	Square	F Value	Pr > F
Model	4	0.00052174	0.00013044	95.05	<.0001
Error	36	0.00004940	0.00000137		
Uncorrected Total	40	0.00057115			

Root MSE	0.00117	R-Square	0.9135
Dependent Mean	0.00332	Adj R-Sq	0.9039
Coeff Var	35.26328		

Parameter Estimates

		Parameter	Standard		
Variable	DF	Estimate	Error	t Value	Pr > |t|
V0	1	0.00044302	0.00010829	4.09	0.0002
V_Target	1	-0.00056309	0.00012261	-4.59	<.0001
V_Tester	1	0.00066984	0.00013092	5.12	<.0001
Mu	1	24.21780	5.18292	4.67	<.0001

Model: Linear_Regression_Model

Dependent Variable: g

Number of Observations Read	60
Number of Observations Used	60

Note: No intercept in model. R-Square is redefined

Analysis of Variance

		Sum of	Mean		
Source	DF	Squares	Square	F Value	Pr > F
Model	4	0.00091121	0.00022780	151.30	<.0001
Error	56	0.00008431	0.00000151		
Uncorrected Total	60	0.00099552			

Root MSE	0.00123	R-Square	0.9153
Dependent Mean	0.00365	Adj R-Sq	0.9093
Coeff Var	33.65212		

Parameter Estimates

		Parameter	Standard		
Variable	DF	Estimate	Error	t Value	Pr > |t|
V0	1	0.00019753	0.00006827	2.89	0.0054
V_Target	1	-0.00023690	0.00008790	-2.70	0.0093
V_Tester	1	0.00038939	0.00009812	3.97	0.0002
Mu	1	31.60173	4.38696	7.20	<.0001

Model: Linear_Regression_Model

Dependent Variable: g

Number of Observations Read	74
Number of Observations Used	74

Note: No intercept in model. R-Square is redefined

Analysis of Variance

		Sum of	Mean		
Source	DF	Squares	Square	F Value	Pr > F
Model	3	0.00146	0.00048744	250.66	<.0001
Error	71	0.00013807	0.00000194		
Uncorrected Total	74	0.00160			

Root MSE	0.00139	R-Square	0.9137
Dependent Mean	0.00416	Adj R-Sq	0.9101
Coeff Var	33.55664		

Parameter Estimates

		Parameter	Standard		
Variable	DF	Estimate	Error	t Value	Pr > |t|
V_Target	1	-0.00014695	0.00006791	-2.16	0.0338
V_Tester	1	0.00064273	0.00007330	8.77	<.0001
Mu	1	26.74336	4.32641	6.18	<.0001

Fig. 18. Multi-variate linear regression fitting on parameter 4g (tetra-mer g parameter). Top left used 40 observation (different combinations of two tester, five targets and four µ values) initially and the model fitted achieved an adjusted R-squared value of 90,39%. Taking a 5% confidence level of significance, only variables V_Tester (Tester internal variance), V_Target (Target internal variance), V0 (Compositional variance difference between tester and target) and µ plays an important factor in explaining parameter 4g. Top right uses 60 observations which contain three testers instead of two achieved adjusted R-squared value of 90.93% with the significant variables remaining the same. Similarly, on the bottom results with 74 observations (removed 6 outliers); the adjusted R-squared value is 91.01% even though one variable V0 is no longer needed within the equation. Hence parameter 4g can be sufficiently modeled by a function of V_Tester, V_Target and µ irrespective of an increasing number of testers based on the adjusted R-squared value.

Based on the results from the previous section, the Verhulst Equation fits well onto the simulated data for different combinations of tester and target with a few outliers as seen in figure 14. From figure 12b and 13, we can also deduce that there is some form of dependence between the characteristics of the tester target sequences with the Verhulst Equation parameters (Figure 18). In this section we proved that there is such dependence and by utilizing this and forming an equation, we can estimate the Verhulst Equation for any combinations of tester and target sequence from bacterial genomes with relative accuracy (Figure 17). This accuracy is affected by the outliers as seen in figure 16 and 17 where by removing the outliers causes a great increase in accuracy of the Verhulst Equation parameter estimate. This Parameter Estimation Function can then be used to estimate the Verhulst Equation which in turn estimates the time of insertion for any genomic island within any recipient genome.

Time of Insertion Estimation

Assuming that for every genomic island insert, there is a unique Verhulst Model explaining the amelioration process of that genomic island. Therefore based on the Verhulst Model, the parameters for each combination of tester and target should also be unique. Hence putting any variable as the subject of the formula with the other variables given, the estimated variable should only have one possible solution. Verhulst Equation uses the function of parameter g and K in order to estimate the iteration used for different composition variance between tester and target. Utilizing the parameter functions of g and m to create K for di-mer, tri-mer and tetra-mer, a time estimation can be made by varying different values of µ. Manipulating equation [6] such that the variable t is the unknown with other variables given, t is then the function as follows:

$$t = \frac{\ln \left| \frac{V_0(K - V(t))}{V(t)(K - V_0)} \right|}{-g} \qquad [9]$$

In equation [9], two parameters are calculated from the GI sequence. V_0, the compositional pattern variance between the donor of the tester sequence and the target and V(t) is the compositional pattern variance between the tester and the target sequence. Equation [7] and [8] are used to calculate g and K according to the regression parameters of the data set. This is done for each K-mer pattern (di-, tri- and tetra-) creating three Verhulst Equations with t being the subject of the formula and resulting in three t values. The µ value which gives the lowest standard deviation measure between the three t values is considered and the time of insertion is equal to the mean of the three t values.

The resulting t value will represent how many iteration it took for the donor sequence (origin of the GI sequence) to reach the state of the GI sequence. A python script is written for this procedure where the µ value used to calculate g and K are rounded off to 6 decimal places. The range of the µ value is determined by the user. For this project specifically, four genomic islands (PAGI1, PAGI2, PAGI3, PAGI4 – PAGI: P.*aeruginosa* Genomic Island) were used with the origin sequence (PKLC 102) being the donor for all four GIs with µ value range of [0.000001; 0.0001].

From the parameter equation [7] and [8] from the method section, we can estimate parameter g, m and K (where K = g/m) when given the variables V_{Target}, V_{Tester}, V_0 and µ. Subsequently with g and K calculated, an amelioration model can be made based on the Verhulst Equation when given V_0 (Compositional variance between donor sequence of GI and target sequence), g, t (number of iterations to achieve composition V(t)) and K. Changing the subject of the formula by letting t be the variable and V(t) (Compositional variance between tester and target sequence at time t) given, we can calculate the time of insertion for any combination of tester and target if we have V_0.

Table 5 shows the results of 4 genomic island inserts PAGI 1, PAGI 2, PAGI 3 and PAGI 4 with the origin sequence for all 4 GIs PKLC and their time of insertion. The table was calculated using excel along with g, k values (di-mer, tri-mer and tetra-mer) from Verhulst Equation fitting of simulation data. Iteration (variable t) value was calculated using equation [9] for different K-mers (2, 3 and 4) for each GI insert. Four different µ values were used which gave different time estimates and each K-mer. The time of insertion estimate for each GI is shown in the last table with label "T" which took the lowest standard deviation measure as the criteria for the four µ values used. The V_0 value between PAGI 2 and origin sequence were shown to be the furthest followed by PAGI 1, 3 and 4 and this is also reflected by the "T" variable where the time of insertion is the most out of all the GIs in the analysis. PAGI 4 which has a higher V(t) than V_0 calculated a negative T value which is correct in the sense of vector signs (negative meaning the opposite direction) showing sequence PKLC requires 42 iterations to achieve the same composition as PAGI 4 or from PAGI 4. We can also say that the corresponding µ value is the true average substitution rate for the GI sequence amelioration process.

Table 5. Time of Insertion Estimation Results

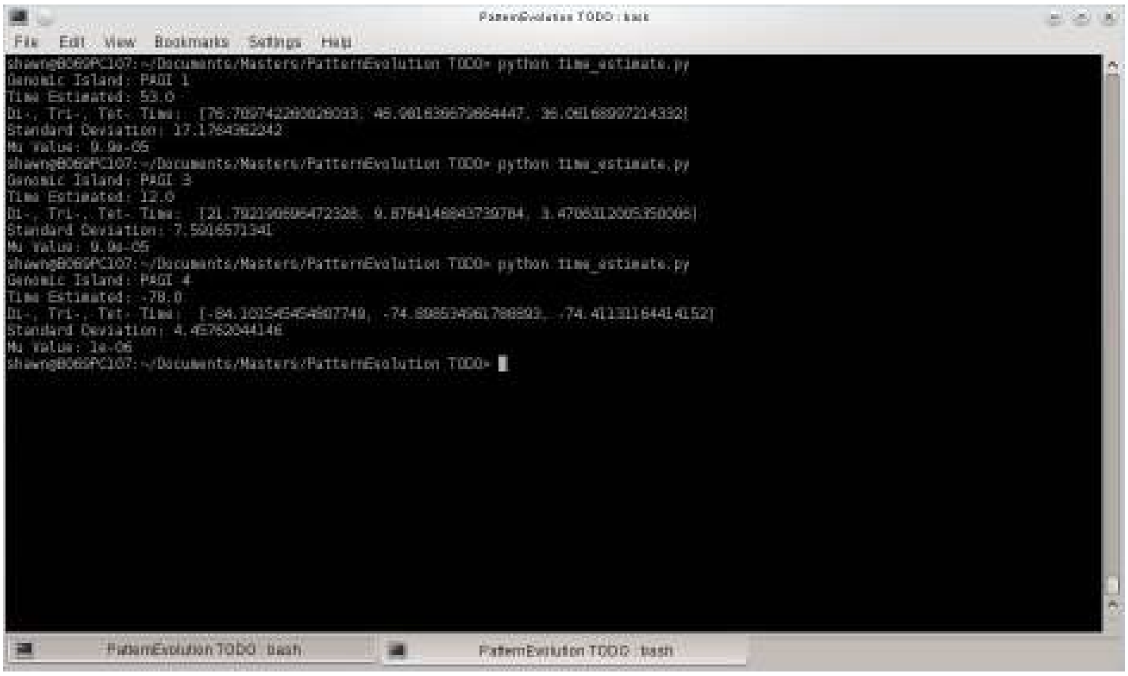
Mu	2K	2g	3K	3g	4K	4g
0.00001	0.99684	0.000346	1.54217	0.000795	1.99905	0.001428
0.00004	0.51	0.000371	1.13	0.001095	1.77	0.002269
0.00008	0.52	0.000539	1.05	0.001353	1.72	0.002889
0.0001	0.4904	0.000585	1.0522	0.001436	1.7003	0.003604

Mu: 0.00001	**2t**	**3t**	**4t**	**Std Dev**
PAGI 1	412.113728	518.566167	720.659124	156.72372
PAGI 2	8931.19813	-	-	-
PAGI 3	117.523322	99.9094962	48.4275954	35.9046245
PAGI 4	-156.295694	-142.671988	-172.120966	14.7381983
Mu: 0.00004	**2t**	**3t**	**4t**	**Std Dev**
PAGI 1	156.979076	195.483716	263.047387	53.6934873
PAGI 2	1405.61648	1858.35832	-	320.136824
PAGI 3	46.1178622	40.6640126	21.0915241	13.1601949
PAGI 4	-63.0218118	-60.5114349	-78.462327	9.72064841
Mu: 0.00008	**2t**	**3t**	**4t**	**Std Dev**
PAGI 1	110.626395	139.204404	187.126677	38.6556164
PAGI 2	997.119789	1100.68528	-	73.2318605
PAGI 3	32.4843438	29.2469093	15.3647134	9.09466957
PAGI 4	-44.3703468	-43.7840766	-57.5998967	7.81282691
Mu: 0.0001	**2t**	**3t**	**4t**	**Std Dev**
PAGI 1	94.9590908	131.62521	144.438802	25.680194
PAGI 2	839.692797	1044.83987	-	145.060883
PAGI 3	27.9237968	27.6473342	11.9609959	9.13736521
PAGI 4	-38.193555	-41.3828649	-44.9685464	3.38942764

	Min Std Dev	**T**	**Mu**
PAGI 1	25.680194	123.674367	0.0001
PAGI 2	73.2318605	1048.90253	0.00008
PAGI 3	9.09466957	25.6986555	0.00008
PAGI 4	3.38942764	-41.5149888	0.0001

2: Di-mer, 3: Tri-mer, 4: Tetra-mer, K and g: Verhulst Equation parameters, T: Time of insertion estimation in terms of iteration, t: estimated iteration based on Verhulst Equation, Std Dev: standard deviation, Mu: µ, PAGI: P.*aeruginosa* Genomic Island.

Instead of using the parameters of the fitted Verhulst Equation which potentially uses a longer time due to simulation process, estimated g and m parameters was done using a python script by varying the µ variable between a range of [0.000001;0.0001] (Figure 19). Utilizing the parameter function and the variables estimated using regression, g and m were calculated and in turn T the same way as Table 5. The estimated time of insertion from the python program show a definite difference to the results from table 5 but the trend in the result remains the same (PAGI 4 being negative, PAGI 1 being more distant than PAGI 3 hence the longer time of insertion). This difference could be caused by the error from the multivariate regression where the model used to estimate the parameters were not 100% (adjusted R-squared) but still serve as a good estimate of the time of insertion. Another interesting point that is shared by both the empirical and theoretical method is the contribution of standard deviation between the three t values (2-mer, 3-mer and 4-mer t estimate). Tri-mer and tetra-mer t estimates are very similar compared to the di-mer t estimate which has a large difference.


PatternEvolution TODO : bash
File
Edit
View
Bookmarks
Settings
Help
PatternEvolution TODO : bash
PatternEvolution TODO : bash

```
shawn@BO6PC107:~/Documents/Masters/PatternEvolution TODO> python time_estimate.py
Genomic Island: PAGI 1
Time Estimated: 53.0
Di-, Tri-, Tet- Time:  [76.709742280028033, 46.981636679864447, 36.081689972143392]
Standard Deviation: 17.1764362242
Mu Value: 9.9e-05
shawn@BO6PC107:~/Documents/Masters/PatternEvolution TODO> python time_estimate.py
Genomic Island: PAGI 3
Time Estimated: 12.0
Di-, Tri-, Tet- Time:  [21.792190696472328, 9.8764148843739784, 3.4706312005350006]
Standard Deviation: 7.5916571341
Mu Value: 9.9e-05
shawn@BO6PC107:~/Documents/Masters/PatternEvolution TODO> python time_estimate.py
Genomic Island: PAGI 4
Time Estimated: -78.0
Di-, Tri-, Tet- Time:  [-84.101545454807749, -74.898534961788893, -74.411311644141527]
Standard Deviation: 4.45762044146
Mu Value: 1e-06
shawn@BO6PC107:~/Documents/Masters/PatternEvolution TODO> 
```

Fig. 19. Python script output for estimation of time of insertion by minimizing standard deviation. Genomic island sequences used were P.*aeruginosa* genomic island (PAGI) 1, 2, 3 and 4 all with origin sequence PKLC. The script output include name of genomic island sequence, time estimated (time of insertion based on minimizing standard deviation), di-mer, tri-mer and tetra-mer time estimate, minimum standard deviation and mu value used to calculated the time of insertion. PAGI 2 was not displayed due to the estimated capacity value were higher than the V_0 difference hence no time was able to be calculated.

There is a significant difference between the simulation data inferred Verhulst Equation and Parameter Estimation Function inferred Verhulst Equation in terms of time of insertion estimate. This is caused by the imperfect parameter estimation function as shown in figure 16 from previous section. But both methods give a good estimation in terms of giving a relative idea to how long ago the genomic island was inserted within the recipient genome (Table 5 and Figure 19). Though the empirical method is more accurate in terms of the lowest standard deviation criteria, this method is very time consuming due to the unknown average mutation value used within the simulation process hence a trial and error approach must be used. On the other hand, the theoretical methods is much faster but less accurate and the average mutation value can be estimated.

Mutation Rate Estimation for Different Genomic Loci

If we view the amelioration process as a change in the mutation rate of the tester sequence within the target throughout time, then can we think of a way to analyze the mutation rate of any genomic loci within the genome sequence? Changing the current algorithm slightly by letting the tester sequence be the genomic loci sequence and target the whole genome sequence, we attempt to estimate the rate of mutation of the genomic loci sequence in terms of differential equation. By analyzing different genomic loci sequences from the same genome sequence by their difference in OU composition, we can analyze the trend between these combinations of tester and target. This trend can be measured in terms of differential and can be viewed as the rate of mutation of the genomic loci sequence relative to other genomic loci of the same genome sequence.

Utilizing the different output parameters from the simulation process, an estimation of the mutation rate of different compositional pattern can be done. Assuming the tester sequence to be a specific sequence of genomic loci and target being the rest of the genome sequence, a rate of mutation measure can be estimated for different genomic loci using a similar approach as the Verhulst Equation with some slight variations to the algorithm. Using the different genomic sequences at different iterations of the simulation process (at each iteration, the composition between the genomic sequence and target differ) as an imitation of different genomic loci, a differential equation representing the mutation rate at each composition pattern variance can be estimated. For this study, three different simulation processes with different μ values (0.00005, 0.0001 and 0.0002) were used to create different sequences with different compositions.

The dataset consist of 60 sequences (2000 iteration simulation each with sequence saved at every 100 iteration period using combination of P.*aeruginosa* genomic island as tester and P.*aeruginosa* whole genome as target). Each sequence represents a theoretical genomic locus (tester) with different composition to the same target. Compositional statistics are then recorded for each combination of tester and target which are sorted in ascending order according to internal variance and then used for the estimation of the differential equation. The variation comes in during the differential equation estimation where the 2 dataset used for the estimation are the composition variance between tester and target and a measure called the forward mutation ratio. The forward mutation ratio is calculated as:

$$Forward\ Mutation\ Ratio = \frac{\text{Distance Between Tester and Target}}{\text{Internal Variance of Tester}}$$

In this study specifically, only the tetra-mer distance and variance is considered and used to calculate this ratio. For each genomic locus, the forward mutation ratio is calculated with the corresponding composition pattern variance between loci and target. Using multiple points from each genomic locus, a differential equation can be calculated using the same method as in figure 10. However the form of the differential equation will differ from equation [5] depending on the dataset given. The resulting differential equation will be an estimate of the mutation rate for any given composition variance between genomic loci and target as represented in equation. This mutation rate measure is a relative measure compared to other genomic loci sequences to how more likely it will mutate compared to the target sequence.

60 sequences simulated with three different μ values were used as a representation of different genomic loci (tester) with the same target (see appendix table 2). Five randomly selected sequences are shown in table 6 with the composition statistic displayed which are needed to calculate the mutation rate for each sequence. The two parameters that determines the mutation rate (differential = mutation rate) are V_0 and forward mutation rate (FMR) and in turn, FMR is calculated from V_{Tester} and distance. Therefore with the three compositional statistics present, a mutation rate estimate can be done for any combination of tester and target.

Analyzing the function (mutation rate is a function of the three compositional statistics) in which the mutation rate is estimated for different genomic sequences, we can determine the relationship of the differential equation based on the three parameters. Looking at table 5 in more detail, three trends can be seen that influences the resulting mutation rate. A decrease in differential can be caused by an increase in V_{Tester}, decrease in distance and decrease in V_0 between tester and target. Thinking in a biological sense, a smaller distance and V_0 between tester and target means that the tester sequence is compositionally similar to the target sequence. Therefore the tester sequence should undergo less mutation due to the sequence reaching a more stable state (higher selection). Hence the relationship between distance, V_{Tester} and mutation rate makes sense here.

Table 6. Mutation rate estimation for 5 simulated sequences

Internal 4V	Distance	V_0	FMR	Differential
5.17	7.67	2.97	1.483559	0.884918
5.63	6.12	2.33	1.087034	0.696544
5.83	5.53	2.1	0.948542	0.62334
6.06	5.5	1.91	0.907591	0.571682
6.28	6.02	1.86	0.958599	0.515892

Internal 4V (V_{Tester}): Tetra-mer internal variance of tester, Distance: Absolute distance measure between tester and target (See section 2.2 in Literature Review), V_0: Variation between composition pattern between tester and target, FMR: Forward Mutation Ratio, Differential: Empirical differential calculated based on FMR and V_0, also equivalent to mutation rate, Number of Mutations: Differential multiplied by loci sequence size.

Figure 20 displays the estimated differential equation estimated from the 60 simulated sequences. The linear differential equation also follows the trend stated above and can be used to estimate the mutation rate based on the V_0 between tester and target. However this estimate is a relatively poor indication due to the non one to one relationship between V_0 and mutation rate (for the same V_0, there could be multiple mutation rate estimates). As seen at the lower left part of the blue dot plot in figure 20, one V_0 measure can result in two different differential values. Hence the differential equation can serve as a trend indicator but not a good estimator of mutation rate. Therefore empirical methods work more accurately than using the differential equation in to estimating the mutation rate.

By creating a new measure FMR in combination with V_0, the change of each individual with respect to the other can be thought to be the change in composition (V_0) with respect to the individual characteristics of the genomic loci sequence (FMR). Hence this measure can be thought as the mutation rate estimate of the genomic loci sequence under study. This algorithm is similar to the Verhulst Equation estimate with the tester and target sequence being the genomic loci sequence and whole genome sequence respectively. The rate of mutation estimate for genomic sequence in this case can only be measure relatively to other genomic loci sequence of the same genome and an increase in the sample number will also increase the accuracy of the mutation rate estimate. Though in theory this measure can be thought of as the rate of mutation estimate, no biological application has been used to prove it being so hence lack the core knowledge to make conclusive statements regarding the use of this algorithm to estimate the rate of mutation for genomic loci.

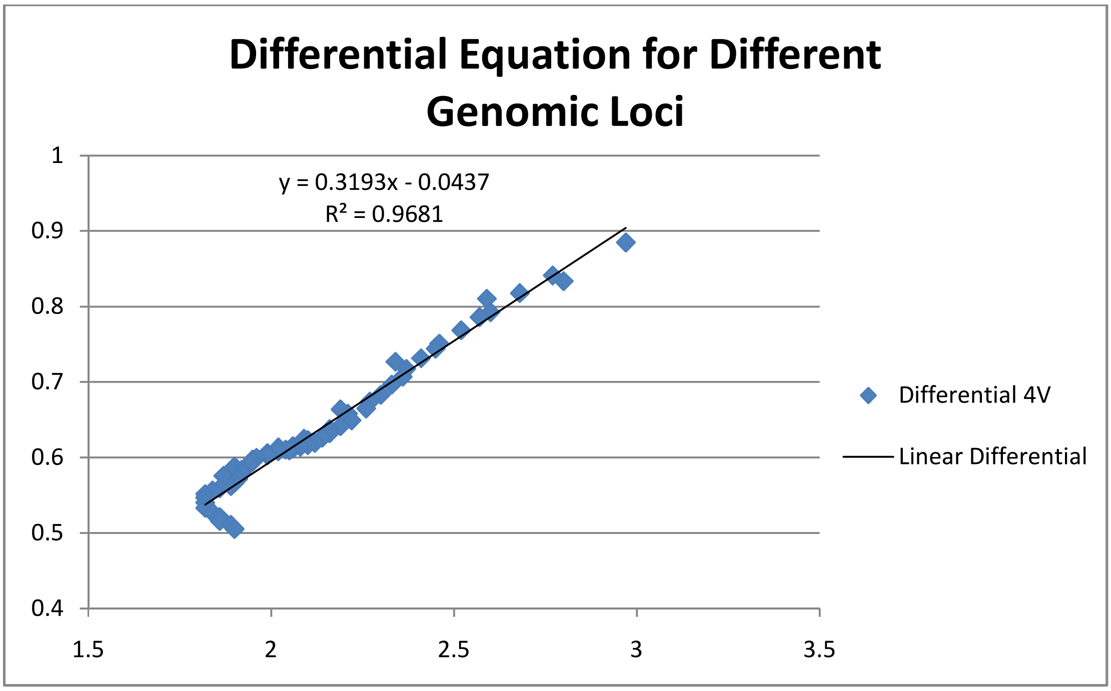


Fig. 20. Empirical Estimation of the differential equation based on the different genomic loci sequences represented by simulation sequences. Each point on graph is plotted by a genome locus (tester) V_0 against the same target on the x-axis with the forward mutation ratio on the y-axis. The differential equation estimated follows a linear function where with any given V_0 between tester and target, the differential is estimated. The differential in this case is equivalent to the mutation rate of the genomic locus with the specific V_0 and forward mutation rate. The linear function is a decreasing function with a decrease in V_0 leading to a decrease in mutation rate. Although the r-squared value is high, the linear function is not a good indicator of the relationship between V_0 and mutation rate. This is caused by the non one to one relationship where one V_0 measure can lead to two mutation rates (caused by different forward mutation rate).

Discussions

Choosing Combinations of Tester and Target

The primary aim of the project is to create an amelioration model which applies to all bacterial genomes. To achieve the maximum accuracy of model, choosing the correct combinations of tester and target is vital. Due to time constraints of the project, five targets and four testers were chosen and each were carefully selected. The five targets were selected first and it is chosen to represent as much of a variety of bacteria as possible. The targets include Bacillus subtilis 168 (BS), Pseudomonas aeruginosa PA01 (PA), Escherichia coli K12 (EC), Xylella fastidiosa 9a5c (XF) and Streptomyces griseus NBRC 13350 (SG). Each target originates from different taxa, found in different environments and has different compositional statistics (GC content, genome size). EC is a gram-negative proteobacteria, most well studied bacteria and found in the human gut. XF is also a proteobacteria and an important plant pathogen which causes phoney peach disease. BS is a gram-positive bacterium of bacilli class which can live in extreme conditions due to its structure. PA is another proteobacteria which is a disease causing bacteria to both animals and humans and can be found in many environments throughout the world. SG is a gram-positive actinobacteria commonly found in the soil with some strains from the deep sea. The GC content varies highly between the targets (EC with 43.51% to SG with 72.2%) with the longest genome size of SG being three times longer than XF (Table 7).

Table 7. Target genome details (UCSC Genome Browser)

Targets	GC Content	Length	Gene Number
B. *subtilis* 168	43.51	4215606	4422
S. *griseus* NBRC 13350	72.2	8545929	2674
X. *fastidiosa* 9a5c	52.67	2679306	2838
E.*coli* K12	50.79	4639675	4466
P. *aeruginosa* PA01	66.56	6264404	5682

The four testers include P. *aeruginosa,* B. *subtilis,* E. *coli* and S. *coelicolor* (SC) which was selected for three purposes. First is to test if the tester and target combination belonging to the same organism will affect the outcome of the Verhulst Equation (e.g. PA and PA). The second is to analyze if closely related organisms will influence the parameters of the model (SG and SC) and lastly if changing the order of tester and target combination will yield the same model. From the resulting Verhulst Equation fitting of all 80 combinations of tester and target, the first and second point does not prove significant (related variable being non-significant). The third point also did not yield the same parameters for different order of the same tester and target combinations (e.g. BS/PA and PA/BS). However the trend in the results does show that there are similarities for these types of combinations as shown in figure 12b and 14.

The parameter estimation result from SAS indicates that there is still room for improvement. Based on figure 18, additional combinations of tester and target should not significantly change the model itself as well as the accuracy in which it predicts the parameters. Hence adding more combinations of tester and target should increase the accuracy of the model even more. But taking into mind the criteria (bacteria taxa, composition and other factors e.g. environment, origin, pathogeneity) in picking these combinations will greatly influence the significance of the model as well as identifying other potential interesting factors which was not found from the current project.

Simulation Parameters

Simulation step takes the longest period of time to run as well as the most important step in structuring the amelioration model. Careful planning needs to be done in order to get the most information out of the simulation data without wasting too much time on computation. Due to the computation complexity of the simulation program, the computation time is exponential when increasing the amount of iterations ran for the simulation. Hence a need to balance the choice of simulation parameters such that maximum information can be kept with the minimum amount of computational time used to calculate it.

Analyzing the 10000 iteration simulation run data with respect to V_0 change per iteration, we want to identify the smallest iteration cutoff point such that there is sufficient information to estimate the parameters of the Verhulst Equation (Figure 21). From the 10000 iteration simulation results, at 2000 iterations, we can identify majority of the change in the curve (slope) which is used to estimate parameter g and also close to the minimum capacity V_0 point which is parameter K. Considering four testers, 5 targets and four μ combinations there will be at least 240 simulations runs. 2000 iteration simulation run takes five days to complete and four simulations can be run simultaneously (adding any more simulations at the same time will affect computation time) therefore it is feasible to do all simulations within a one year period (estimated time taken for all simulations is 300 days).

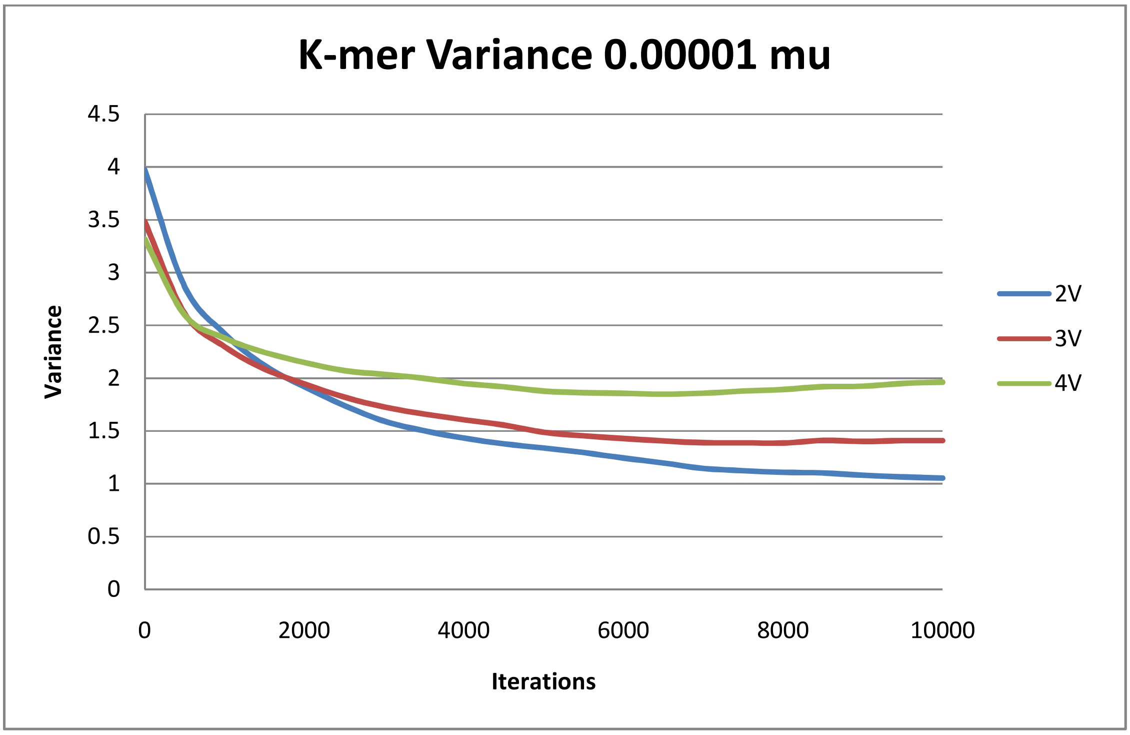


Fig. 21. 10000 iteration simulation V_0 graph for different K-mers at 0.00001 µ. Majority of the decrease occurs before 2000 iterations and then start to tend to equilibrium. Hence by adding iterations more than 2000 will not greatly increase the information needed to estimate the parameters (g – slope of the curve and k – Minimum limit V_0 value) for the amelioration model.

Four µ values were chosen such that the µ values within the specified range do not follow a specific trend (linear) and still able to determine if µ is a factor in estimating the parameters of the Verhulst Equation. From the Verhulst modeling and parameter estimation results, the µ values used were significant in determining the parameters but were poor in fitting the model to the simulation data. From the box and whisker plot of the residue differentiated by the different µ values (Figure 12a), the lowest µ value used (0.00001) were not a good choice as a simulation parameter. This could however be compensated by increasing the number of iterations used within the simulation to get more data for estimation or just use a higher µ value. A possible improvement to the current existing model could be done by changing the simulation parameters (increasing iteration count, increasing µ value).

Probability Logistic Function

In order to construct a sensible amelioration model, the simulation data must be as close to a true amelioration process as possible. The Probability Logistic Function (PLF) is especially tailored for the simulation process to do this with two assumptions in mind which makes this function biologically suitable. The first assumption is the most important as well as the core idea of amelioration which is directional mutation. Based on literature review section 3.1, Lawrence and Ochman (1997) proved with their amelioration model that directional mutation occurs and drives the amelioration process. The simulation process utilizes this core concept and changes the compositional statistics of tester and target as a measure of probability of substitution at each base position. Meaning that depending on where the sequences (OU pattern) between tester and target are different, a likely substitution at that point will be a high in probability. Therefore the assumption of directional mutation is the fundamental core of the simulation process.

Second assumption is that every bacterial genome should differ in some way and hence the way they ameliorate should differ. Hence the PLF should be able to adjust itself depending on the biological fact of the sequence. This is done with parameters “a” and “b” where parameter “a” is a function of µ which directly controls the likelihood of substitution and “b” which controls the conservations of the sequence. Though “a” in this case is a constant mutation rate where under no directed mutation or selection occurs, the mutation probability or likelihood of mutation is never constant for any base position of the tester sequence due to mutation pressure of the target genome. This is an important consideration as some sequences tend to have a different mutation rate than other regions. Using different combinations of “a” and “b” will give you a unique representation of different amelioration process simulations for different combinations of tester and target. This will also give more flexibility in which the function can be manipulated to represent as closely to real life applications.

For the 80 combinations of tester and target combination simulations, “a” parameter was being varied by the change of µ with “b” parameter all equal to 1. Based on the SAS results from figure 16, the adjusted R-squared was very low and was a poor fit. This poor fitting was caused by the outliers from specific combinations of tester and target. Looking at it from another perspective, these outliers could potentially be caused by the incorrect usage of parameter “b” and hence create bias within the simulation step. Therefore, the correct choice of parameter “b” could increase the accuracy of the amelioration model as well as the parameter estimation of the model. For future research, the connection between parameter “b” and the Verhulst model could enhance the biological significance of the model.

Verhulst Equation

Overall, the fitting of the Verhulst Equation on the simulation data of 80 combinations of tester and target was on average a good fit. Due to the nature of the amelioration process (directional mutation), the logistic curve fit well to the simulated data. Some combinations were shown extremely good fits where the residue value between the actual simulation data and estimated model was 0.4003 (absolute measure of difference). However some shown to be not compatible with the Verhulst Equation on two reasons with the first being the actual simulation data. When the composition of the tester and target differ significantly (e.g. BS/SG and SC/BS combinations), the simulation data becomes extreme which result in high minimum V_0 value and extreme difference between initial V_0 and minimum V_0 (causes extreme slope hence high parameter g). Therefore when fitting a Verhulst Equation with irregular parameters K and g (as seen in figure 15 with the outliers) which forms the core of the shape of the equation, the residue value will naturally be high which proves it to be a none good fit. The second reason is the estimation of the differential equation where the fitting of the Verhulst Equation is strictly dependant on. However the differential equation is also derived from the simulation data hence the first reason will also apply here. The actual estimation of the differential equation works by transforming the differential dataset into a linear function and then estimating both g and K (see methods). In order to estimate a sensible K value from the linear function, all values which are close to equilibrium are removed and a cutoff point is set. To get this cutoff point, two criteria were considered and only one was used in the python script. The first criteria was that all points lower than a certain percentage change from the previous iteration was set as the cutoff while the second criteria which was used within the python script sets the cutoff at the point when the differential is positive.

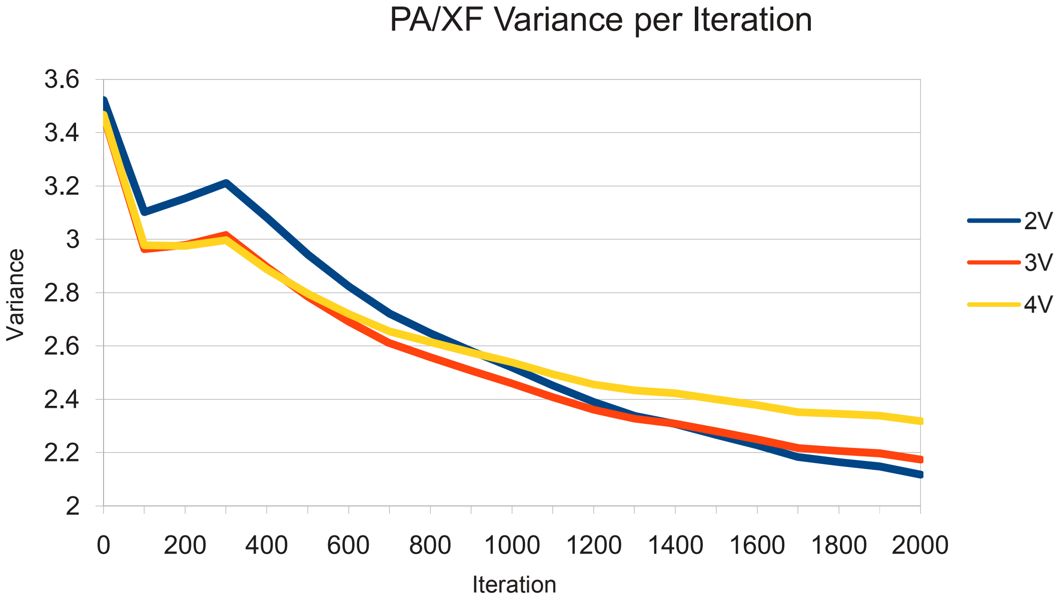


Fig. 22. Simulated amelioration process of combination P.*aeruginosa* tester and X.*Fastidiosa* strain 9a5c target. 2000 iteration and μ value of 0.00001 was set as the input parameter. At 300 iterations, a “mountain peak” structure occurs which is caused by a substitution away from target instead of towards it. This causes a shift in the cutoff value being at 300 iterations instead of 2000 hence the K value estimation will be around 2.9 instead of 2.1. Therefore setting the count number of differential sign changes (change from increase – positive to decrease – negative or vice versa) to 2 in this case will prevent the premature setting of the cutoff. Hence investigating the simulation data before hand and manually setting the criteria will improve the accuracy of the Verhulst Equation fitting to simulation data.

Looking at both criteria, there are certain situations when the simulation data will create flaws when estimating parameter K. For the first criteria, if the composition of both tester and target are relatively close (e.g.EC/EC combination) then the percentage change per iteration will naturally be very low (V_0 per iteration change of less than 0.5%), hence this will cause the K estimation to be higher than it actually is (cutoff set too early). For the second criteria, in rare circumstances, some combinations show irregular change in their V_0 per iteration such as a “mountain range” structure (Figure 22). This structure means that within the amelioration process, at some iteration the change of V_0 is moving away from the target instead of towards it. Therefore the shape of the amelioration process will not be a continuous decreasing graph but an increasing and decreasing graph like the shape of a mountain range. In this case, the cutoff point will also be set too early and hence the estimation of parameter K will be bigger than it should be. In order to avoid these circumstances, manual inspection of the simulation takes priority in the sense of knowing what type of data you are dealing with. Changing the criteria to suit the simulation data under study such as increasing the count number in which the differential is positive (count number 2 was used within study) in the second case and for the first case, you can use a smaller percentage change to match the simulation data. Other changes which could improve the Verhulst Equation fitting would be the usage of parameter "a" and "b" for different combinations of tester and target. From figure 12a, a change in parameter "a" improved the fitting of the Verhulst Equation for the same tester and target combination.

Parameter Estimation and Selection Methods

Taking into account all combinations of tester and target data and possible influence factors which contribute to the estimation of the parameters, the estimated model fit the data poorly as seen in figure 16. The core philosophy of the amelioration model was to create a simplistic equation from simulation data that applies to all bacterial genomes. If data were to be eliminated to get a better fitting as in figure 17, then the model itself is not a reflection of what is intended from the aim and the model itself is biased. Hence improvement needs to be done in order to achieve the results of figure 17 without the removal of the extreme combinations which causes the bad fitting of the parameter function.

Two core problems can be seen from the above problem with the first being the cause of the extreme outliers and the second is the variable selection of the model. Through the discussion of "Verhulst Equation" and "Probability Logistic Equation" section, the first problem can be reduced through the control of the simulation and fitting process. Second problem can be seen by comparing figure 16 and 17. If you consider the variable to be significant at 5% confidence level, then in figure 16 the variables that are significant are V0, V_Tester and V_Target while in figure 17, variables V_Tester, V_Target and Mu are significant. This example implies two points with the first being the variables considered for the model will influence the effects of other model (collinearity effect). For instance in figure 16, six variables are considered hence the variable V0 effect in the model overtake the effect of Mu therefore Mu is less significant than V0. However in figure 17 after non-significant variables are removed, the effect of Mu is more apparent than when there was 6 variables in figure 16. This problem can be easily solved through different selection methods to choose the best model suitable for the data under study.

Three selection methods were used in the study and the stepwise selection methods was chosen and used for all parameter rather than the other two selection methods on two reasons. Stepwise selection methods take less computation time (Though in this study, computation time does not apply due to the small number of variables used) and are a combination of both forward and backward selection method. By setting an entry and exit level of significance for variables under consideration for the model, stepwise selection method takes a "jumping" approach. The method start off with analyzing each variable and according to the entry level of significance, it will accept the variable within the model. While accepting each variable, it will also calculate the contribution of the variable to the model and select the most significant variable. At each step, the significance of the other variables is calculated according to the already selected variables and any non-significant variables are removed according to the exit level of significance. In this way the first core problem stated from the above example can be cleared.

Another issue with the above problem is choosing the best level of significance that is needed to be considered for the variable to be used within the model. From figure 16, the hypothesis was that related tester and target should be a factor influencing the parameter of the Verhulst Equation (PA/PA combination had a very low K value). This idea is reinforced in figure 17 where the level of significance of the related variable is at 7.38%. For this study specifically, a 5% level of confidence is used as both entry and exit level but with a difference of 2.38%, can we disregard the related variable as insignificant to the model? Hence the question of setting the level of confidence can be debatable. However 5% level of confidence was still chosen as a strict criterion to reduce bias from large number of variables used (adjusted R-squared value). In terms of the related variable example, from the parameter estimation function estimated from SAS in figure 17, 0.37% partial R-squared value was lost due to elimination of the related variable. The significance of 0.37% relative to the whole model (91.37% R-squared value) is very low but the significance cannot be regarded and can be considered contextual based. Many other potential variables which could be of interest were not considered in this study such as sequence size, sequence region (Nakamura et al. 2004) and multiple average mutation rate (Snir, 2014). These can be tested in future to further enhance current model for parameter estimation.

Time of insertion estimation

In order to get a sensible measure of time of insertion, a criterion is set for two reasons. For any genomic island (tester) insert at any point in time, the tester sequence can be expressed as compositional characteristics (OU pattern of 2-mer, 3-mer and 4-mer). Hence with these statistics and using the Verhulst Model, time estimation can be done for the tester in question. In the perfect theoretical word where the amelioration model is perfect, the estimated time for all three patterns (2, 3 and 4-mers) should be exactly identical and therefore the time of insertion should be the estimated time value. But in practice, there is a discrepancy between the three estimated time values and this is caused by two reasons. The first is random noise substitution where biological data does not always follow the strict rules of a mathematical model. The second is the imperfect model where the model itself cannot accurately measure the variable in question (not 100% R-squared regression fitting, high residue during Verhulst Model fitting). Therefore the lowest possible deviation between the three time estimates is considered as the set criterion for estimating the time of insertion. However analyzing the standard deviation measure yielded interesting result where the contribution of standard deviation from the three time estimates is shown to be quite different. Tri-mer and tetra-mer time estimate was similar compared to the di-mer estimate which was significantly different. This difference could be caused by the poor Verhulst Model fitting on the di-mer dataset which resulted in the biased t estimate. Hence possible improvement to the current method is by applying a weighting model (give tetra-mer and tri-mer t estimate a higher percentage weighting instead of the average weighting which is used currently in this study) or add higher k-mers into consideration (more t estimates to get a clearer idea of the true time of insertion), the time of insertion estimate might be more accurate.

The second reason behind the set criterion is to assure that the estimated time value is in fact the most correct estimate. To do this, the analysis of the relationship between μ and standard deviation is needed. In order to identify the most correct estimate, there should only be one minimum standard deviation value for all μ values (Figure 23). Consider the opposite where there are multiple low points of standard deviation values, there is no specific way in telling which time estimate is true for the tester sequence in question. Hence the criterion of lowest standard deviation will not be able to estimate the most correct time of insertion. Therefore the criterion set is sufficient in identifying the most accurate time of insertion based on the Verhulst Model.

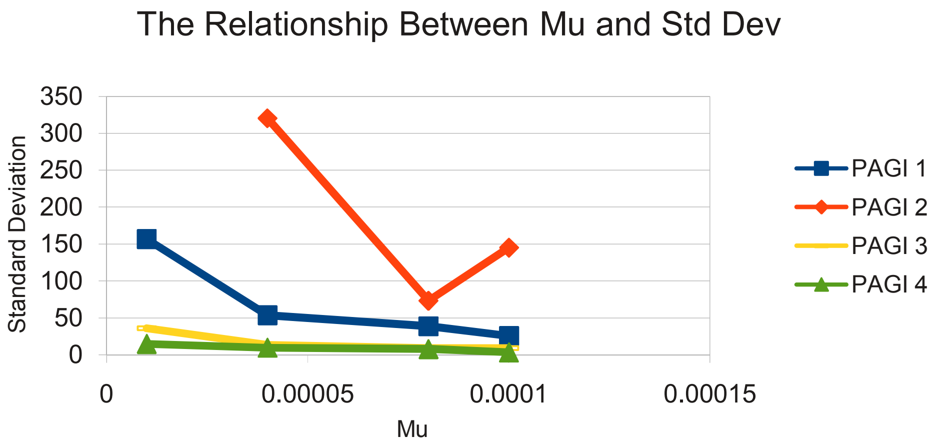


Fig. 23. Relationship between the change in Mu and its effect on the standard deviation measure of the set [2t, 3t, 4t]. From the red curve, we can deduce a parabola curve where the others are more of a linear decreasing curve. Based on the trend, a change in mu will vary the standard deviation value such that at some value of mu, the standard deviation will achieve a minimum value (Red Curve).

Comparing the empirical estimation of the time of insertion and the theoretical approach using the parameter estimation method based on table 5 and figure 19, there is still a significant difference in the estimation. The poor estimation of the theoretical approach is largely responsible by the bad estimations of the parameters using the parameter estimation functions from SAS (Low R-squared values on parameter estimation models). This estimation could however be improved if the parameter estimation is done individually on the combination instead of all possible combinations (specific combinations does not follow the trend of the parameter equation e.g. PA/PA combination has much lower g and K values empirically compared to theoretical estimate). Although the empirical estimation is better in estimating the time of insertion for the simulated sequences compared to theoretical approaches, it is very time consuming in both simulation and calculation step while the theoretical approach is much faster in the sense of given a few input parameters for an output. Hence a better parameter function is needed which was explained in earlier discussions.

There are also two more limitations to the time of insert estimation method. The first is the need of a point of origin sequence for the calculations to work. Within the given parameters of the Verhulst equation, the parameter V_0 is needed to calculate the time variable. Although the origin sequence can be thought as the donor of the tester sequence but sometimes this is not always the case. The tester sequence could be donated from any organism with such sequence hence the choice of which sequence to choose as the origin sequence can largely affect the time estimate. The other limitation is the inability to calculate a time estimate based on the Verhulst Model as seen in table 5 and figure 19 (PAGI 2). This is caused by the simulation step where no simulations could cover the amelioration process of the tester sequence therefore the Verhulst Model cannot estimate the time of insertion for that tester sequence. To solve this problem, simulation step needs to be improved as discussed in earlier section. Ultimately, the time of insert estimation method is a good approach in getting an accurate time estimate but more test data is needed to test its practical uses on real life applications.

There is one point of interest is that of the four GIs analyzed here, there are two different rates of mutation even though they all come from the same origin within the same target. This could potentially mean that there might be more than one possible Mu value for any given tester sequence or that there are more than one model of selection within the target with different stress level mutagenesis (Maclean, 2013). With this thought in mind, this could mean that within the amelioration model, there might be two or more mu parameters which also correspond to another study done by Snir (2014).

Rate of Mutation Estimation

A study by Martincorena (2012) states that there is a great variation in mutation rate across the genome which is non-random and depends on factors (DNA repair, transcription) which reduces risk of deleterious mutations. Codon positions also undergo different mutation rate (Knight 2001), hence combining the two there should be some connection between sequence patterns and rate of mutation. In attempt to find the pattern, three parameters distance (D), internal variance (Int V) and variance (V_0) was used in attempt to characterize a sequence in a unique way such that for each sequence, there is a unique measure of mutation rate corresponding to that sequence. Analyzing each parameter individually, internal variance has the least impact in determining the mutation rate since the internal variance measure only applies to the tester sequence. This parameter serves as a normalizing constant for different genomic loci within the study such that it creates a third dimension to the calculations as well as maintain the unique characterization of each sequence (some genomic loci sequences has got the same distance and V_0 measures to the target but contain completely different internal variance). The distance parameter is a measure between tester and target calculated as the absolute distance between ranks of OU in the two patterns. Hence by normalizing the distance parameter by the internal variance gives you a relative ratio (Forward Mutation Ratio) of how close in terms of distance is the sequence with the specific internal variance is to the target sequence.

The forward mutation ratio (FMR) makes biological sense in two ways. The first case is that lower distance value implies the tester sequence and the target composition are relatively similar. This is equivalent to a low FMR value due to the lower distance and hence lower mutation rate. The second case is where two tester sequences having the same distance measure with different internal variance where the higher internal variance sequence will have a lower FMR value. In such case, the higher internal variance sequence will always be closer in composition to the target than the lesser one. Therefore for each sequence there is a unique FMR measure which is relative to the mutation rate of that sequence (low FMR = low mutation rate).

V_0 is determined by the variability of word deviations between the tester sequence and target. It can be seen as how different is the tester and target sequence in terms of their composition. Assuming that V_0 is the difference between the tester sequence and the target sequence and the FMR value determines the change needed between the two sequences, the differential between the two variables should equal to the mutation rate. Tetra-mer distance and V_0 were chosen as the calculation parameters within this study. The choice of tetra-mer over di-mer and tri-mer was due to the complexity of the tetra-mer pattern (256 word combinations over 16 and 81 respectively) to better reflect the sequence structure. But this assumption was not proven due to the lack of knowledge to determine which mutation rate estimate is more correct compared to others. Hence more in depth study is needed on different K-mer parameters (D, V, Int V) to determine the best approach and also real data (genomic loci sequence instead of simulated sequences) to test the practicality of the method (Comparing the number of mutation to biological sequence data).

The main downside to the approach of estimating the rate of mutation is that it uses a relative approach. Multiple genomic loci sequences (preferably more than five sequences) with the same target (whole genome sequence containing the genomic loci sequence) are required for the method to work. Increasing the number of genomic loci sequences within the calculation will increase the accuracy of the differential measure (Table 8). This is due to the differential being a function of the sample data where the trend of the data determines the differential measure. An advantage to increasing the number of genomic loci sequences is that for each genomic locus, the associated differential will form a differential equation which can be used for comparison between organisms (mutation rate function between organisms) or in depth analysis of the trend within the organism (linear or non-linear relationship between mutation rate and sequence composition). But this causes unnecessary calculations if you are only interested in one or two genomic loci sequences and their mutation rates with regards to the target sequence. Though a measure is calculated based on compositional statistics and is comparative to other genomic loci sequences, the lack of biological data to back up this measure will give inconclusive results.

Table 8. Comparison between 5 genomic loci differential measure using different total numbers of genomic loci used

Internal 4V	Distance	V_0	FMR	Differential N = 5	Differential N = 60
5.17	7.67	2.97	1.483559	0.884918069	0.884918069
5.63	6.12	2.33	1.087034	0.699732181	0.696544331
5.83	5.53	2.1	0.948542	0.677820406	0.623340175
6.06	5.5	1.91	0.907591	0.633470643	0.571681541
6.28	6.02	1.86	0.958599	0.590850028	0.5158925

Internal 4V: Tetra-mer internal variance of tester, Distance: Absolute distance measure between tester and target (See section 2.2 in Literature Review), V_0: Variation between composition pattern between tester and target, FMR: Forward Mutation Ratio, Differential: Empirical differential calculated based on FMR and V_0, also equivalent to mutation rate, N: number of genomic loci sequence used for differential calculation

Conclusion

The amelioration process of a genomic island insert (tester) in a recipient genome (target) is successfully represented through simulation using the probability logistic function (PLF) and the compositional characteristics of the tester and target. Utilizing the correct input parameters which characterizes the tester and target genome (PLF parameter “a” and “b” controlling the mutability of sequence and its conservativeness) allows the simulation to better reflect the true amelioration process of the tester within specific targets. The amelioration model which takes the form of a Verhulst Equation has proven to fit well to the simulation data sets within this study. Hence by knowing the parameters of the Verhulst Equation, the amelioration process can be expressed as a simple equation which can be used for further analysis such as time of insert estimation or amelioration model comparisons between different tester and target combinations.

Parameter estimation model which was estimated through regression from 80 different combinations of tester and target was done in attempt to estimate the parameters of the Verhulst Model for any combinations of tester and target. Although the estimated parameter function is not 100% accurate but it is still within an acceptable range for it to make sensible estimations (parameter estimation model r-squared values all above 70%). The time of insert estimations done through the estimated parameter showed a significant difference to the simulation and model fitting approach but the trend in the answers remains the same. Hence the parameter estimation method is a good way to get an estimated result in a short period of time (only the parameter equation is needed compared to empirical methods needing the simulation data which potentially requires a long time). But, there is room for improvement in all stages of the method (e.g. changing input parameters of PLF to better suit the tester and target for better simulation results, model selection technique during regression of parameter estimation function) where each improvement can increase the accuracy of the of the resulting estimate.

Altering the methods for the estimation of the mutation rate of genomic loci sequence has shown to have not so significant results. Through the usage of the simulations sequences as genomic loci, relative approach was used in attempt to estimate the mutation rate of each sequence. Although sensible mutation rate measures were estimated (in terms of trend between sequence composition and estimated mutation rate), there was no biological significance to the measure in which the mutation rate made enough sense to use it to make any form of conclusions. Therefore more research is needed here to bridge the result with biological data to make further assumption and could potentially result in interesting outcomes.

Acknowledgement

I would like to thank all the members at the Bioinformatics and Computational Biology Unit at the University of Pretoria for all their support, guidance and assistance. In particular my supervisor Dr. O. Reva and Prof. F. Joubert, I would like to thank both of you for everything you have done. Also, I would like to thank NRF and University of Pretoria for their funding, without them, this would not be possible.

References

1. SWGB mirror site at the University of Pretoria in South Africa.

2. BABIC, A., LINDNER, A., VULIC, M., STEWART, E. & RADMAN, M. et al. 2008. Direct visualization of horizontal gene transfer. *Science* 319, 1533–1536.

3. BEIKO, R. & HAMILTON, N. 2006. Phylogenetic identification of lateral genetic transfer events. *BMC Evol Biol,* 6**,** 15.

4. BEZUIDT, O., MENDEZ, G. & REVA, O. 2009. SEQWord Gene Island Sniffer: a program to study the lateral genetic exchange among bacteria. *World Academy of Science, Engineering and Technology* 58, 1169-11274

5. BEZUIDT, O., GANESAN, H., LABUSCHANGE, P. EMMETT, W., PIERNEEF, R & REVA, O. 2011. Linguistic Approaches for Annotation, Visualization and Comparison of Prokaryotic Genomes and Environmental Sequences. In N-Y Yang (Ed) Systems and Computational Biology - Molecular and Cellular Experimental Systems Chapter 2, 27-52. *InTeck, Croatia*. (ISBN 978-953-307-280-7)

6. BOUCHER, Y., DOUADY, C., PAPKE, R., WALSH, D., BOUDREAU, M., NESBO, C., CASE, R. & DOOLITTLE, W. 2003. Lateral gene transfer and the origins of prokaryotic groups. *Annu Rev Genet,* 37**,** 283 - 328.

7. BREIMAN, L. 2001. Random Forests. *Machine Learning,* 45**,** 5 - 32.

8. BUCHRIESER, C., PRENTICE, M. & CARNIEL, E. 1998. The 102-kilobases unstable region of Yersinia pestis comprises a high-pathogenicity island linked to a pigmentation segment which undergoes internal rearrangement. *J Bacteriol* 180: 2321–2329.

9. CHATTERJEE, R., CHAUDHURI, K. & CHAUDHURI, P. 2008. On detection and assessment of statistical significance of Genomic Islands. *BMC Genomics,* 9**,** 150.

10. DAGAN, T., RANDRUP, Y. & MARTIN, W. 2008. Modular networks and cumulative impact of lateral transfer in prokaryote genome evolution. *Proc Natl Acad Sci USA*. 105, 10 039–10 044.

11. FOERSTNER, K., VON, M., HOOPER, S. & BORK, P. 2005. Environments shape the nucleotide composition of genomes. *EMBO Rep,* 6**,** 1208 - 1213.

12. GAL-MOR, O. & FINLAY, B. 2006. Pathogenicity islands: a molecular toolbox for bacterial virulence. *Cell. Microbiol*. 8: 1707–1719

13. GANESAN, H., RAKITIANSKAIA, A., DAVENPORT, C., TUMMLER, B. & REVA, O. 2008. The SeqWord Genome Browser: an online tool for the identification and visualization of atypical regions of bacterial genomes through oligonucleotide usage. *BMC Bioinformatics,* 9**,** 333.

14. GROISMAN, EA. & OCHMAN, H. 1996. Pathogenicity islands: bacterial evolution in quantum leaps. *Cell*. 1996;87:791–794.

15. HACKER, J. & CARNIEL, E. 2001. Ecological fitness, genomic islands and bacterial pathogenicity. *EMBO Rep*. 2001 May 15; 2(5): 376-381

16. HAMADY, M., BETTERTON, M. & KNIGHT, R. 2006. Using the nucleotide substitution rate matrix to detect horizontal gene transfer. *BMC Bioinformatics,* 7**,** 476.

17. HEIN, J., JIANG, T., WANG, L. & ZHANG, K. 1996. On the complexity of comparing evolutionary trees. *Discrete Applied Mathematics* 1996,71:153-169.

18. HOROWITZ, J., NORMAND, M. D., CORRADINI, M. G. & PELEG, M. 2010. Probabilistic Model of Microbial Cell Growth, Division, and Mortality. *Applied and Environmental Microbiology,* 76**,** 230-242.

19. HUSON, DH. & BRYANT, D. 2006. Application of phylogenetic networks in evolutionary studies. *Mol Biol Evol*. 2006;23:254–267.

20. JAIN, R., RIVERA, M. & LAKE, J. 1999. Horizontal gene transfer among genomes: the complexity hypothesis. *Proc Natl Acad Sci USA,* 96**,** 3801 - 3806.

21. KANHERE, A. & VINGRON, M. 2009. Horizontal Gene Transfers in prokaryotes show differential preferences for metabolic and translational genes. *BMC Evol Biol,* 9**,** 9.

22. KARLIN, S. & BURGE, C. 1995. Dinucleotide relative abundance extremes: a genomic signature. *Trends Genet,* 11**,** 283 - 290.

23. KLOCKGETHER, J., WURDEMANN, D., REVA, O., WIEHLMANN, L. & TUMMLER, B. 2007. Diversity of the abundant pKLC102/PAGI-2 family of genomic islands in Pseudomonas aeruginosa. *J Bacteriol*, 2007, 189:2443-2459.

24. KNIGHT, RD., STEPHEN, J. & LANDWEBER, L., 2001. A simple model based on mutation and selection explains trends in codon and amino-acid usage and GC composition within and across genomes. *Genome Biol*, 2, R10.

25. KOONIN, E., MAKAROVA, KS. & ARAVIND, L. 2001. Horizontal gene transfer in Prokaryotes. Quantification and classification. *Annu. Rev. Microbiol*. 55, 709–742.

26. KOSEKI, S. & NONAKA, J. 2012. Alternative Approach To Modeling Bacterial Lag Time, Using Logistic Regression as a Function of Time, Temperature, pH, and Sodium Chloride Concentration. *Applied and Environmental Microbiology,* 78**,** 6103-6112.

27. KOSKI, L. B., MORTON, R. A. & GOLDING, G. B. 2001. Codon bias and base composition are poor indicators of horizontally transferred genes. *Mol Biol Evol,* 18**,** 404-12.

28. LAWRENCE, J. & OCHMAN, H. 1997. Amelioration of bacterial genomes: rates of change and exchange. *J Mol Evol,* 44**,** 383 - 397.

29. LEDERBERG, J. & TATUM, EL. 1946. Gene recombination in Escherichia Coli. *Nature*, 158, 558.

30. LEVINGS, RS., LIGHTFOOT, D., PARTRIDGE, S., HALL, R. & DJORDJEVIC, S. 2005. The Genomic Island SGI1, Containing the Multiple Antibiotic Resistance Region of Salmonella enterica Serovar Typhimurium DT104 or Variants of It, Is Widely Distributed in Other S. enterica Serovars. *J. Bacteriol*. vol. 187 no. 13 4401-4409

31. LOBRY, J. & LOBRY, C. 1999. Evolution of DNA base composition under no-strand-bias conditions when the substitution rates are not constant. *Mol Biol Evol,* 16**,** 719 - 723.

32. MACLEAN, R.C., TORRES-BARCELO, C. & MOXON, R., 2013. Evaluating evolutionary models of stress-induced mutagenesis in bacteria. *Nature Reviews Genetics*, 14(3), pp.221–227.

33. MARTINCORENA, I., SESHASAYEE, A. & LUSCOMBE, N. 2012. Evidence of non-random mutation rates suggests an evolutionary risk management strategy. , 14, pp.6–9.

34. MEDINI, D., DONATI, C., TETTELIN, H., MASIGNANI, V. & RAPPUOLI, R. 2005. The microbial pan-genome. *Curr Opin Genet Dev*. 2005 Dec;15(6):589-94.

35. MOZHAYSKIY, V. & TAGKOPOULOS, I., 2012. Horizontal gene transfer dynamics and distribution of fitness effects during microbial in silico evolution. *BMC bioinformatics*, 13(10), p.13.

36. NAKAMURA, Y., ITOH, T., MATSUDA, H. & GOJOBORI, T. 2004. Biased biological functions of horizontally transferred genes in prokaryotic genomes. *Nat Genet,* 36**,** 760 - 766.

37. OCHMAN, H., LAWRENCE, J. & GROISMAN, E. 2000. Lateral gene transfer and the nature of bacterial innovation. *Nature,* 405**,** 299 - 304.

38. PENN, K., JENKINS, C., NETT, M., UDWARY, DW., GONTANG, EA., MCGLINCHEY, RP., FOSTER, B., LAPIDUS, A., PODELL, S., ALLEN, EE., MOORE, BS. & JENSEN, PR. 2009. Genomic islands link secondary metabolism to functional adaptation in marine Actinobacteria. *ISME J.*2009;3:1193.

39. PODELL, S. & GAASTERLAND, T. 2007. DarkHorse: a method for genome-wide prediction of horizontal gene transfer. *Genome Biol,* 8**,** R16.

40. RAGAN, M., HARLOW, T. & BEIKO, R. 2006. Do different surrogate methods detect lateral genetic transfer events of different relative ages? *Trends Microbiol,* 14**,** 4 - 8.

41. RANDLES, R. 1979. Introduction to the Theory of Nonparametric Statistics.

42. SANTOS, S. & OCHMAN, H. 2004. Identification and phylogenetic sorting of bacterial lineages with universally conserved genes and proteins. *Environ Microbiol,* 6**,** 754 - 759.

43. SMITH, A., LUI, T. & TILLIER, E. 2004. Empirical models for substitution in ribosomal RNA. *Mol Biol Evol,* 21**,** 419 - 427.

44. SNIR, S. 2014. On the number of genomic pacemakers: a geometric approach. *Algorithms for Molecular Biology*, 9(1)

45. SUEOKA, N. 1962. On the genetic basis of variation and heterogeneity of DNA base composition. *Proc Natl Acad Sci USA,* 48**,** 582 - 592.

46. SUEOKA, N. 1988. Directional mutation pressure and neutral molecular evolution. *Proc Natl Acad Sci USA,* 85**,** 2653 - 2657.

47. SYVANEN, M. 1985. Cross-Species gene transfer: implications for a new theory of evolution. *F. Theor. Biol.* 112,333-343.

48. TAMAMES, J. & MOYA, A. 2008. Estimating the extent of horizontal gene transfer in metagenomic sequences. *BMC Genomics,* 9**,** 136.

49. TEMPLETON, AR., CRANDALL, KA. & SING, CF. 1992. A cladistic analysis of phenotypic associations with haplotypes inferred from restriction endonuclease mapping and DNA sequence data. III. Cladogram estimation. *Genetics* 132:619–633.

50. THIERGART, T., LANDAN, G. & MARTIN, WF. 2014. Concatenated alignments and the case of the disappearing tree. *BMC Evol Biol*, 14(1), p.2624

51. THOMPSON, CC., CHIMETTO, L., EDWARDS, R., SWINGS, J., STACKEBRANDT, E. & THOMPSON, F. 2013. Microbial genomic taxonomy. *BMC genomics*, 14, p.913

52. VOGAN, A. & HIGGS, P. 2011. The advantages and disadvantages of horizontal gene transfer and the emergence of the first species. *Biology Direct,* 6**,** 1.

53. WELLNER, A., LURIE, M. & GOPHNA, U. 2007. Complexity, connectivity, and duplicability as barriers to lateral gene transfer. *Genome Biol,* 8**,** R156.

54. ZINDER, N.D. & LEDERBERG, J. 1952. Genetic Exchange in Salmonella. *F. Bacteriol*. 64, 678-699.

Appendix

Table 1. Verhulst Equation Fitting Results (ALL)

		Dimer		Trimer		Tetramer	
Tester/Target	**Mu**	**m**	**g**	**m**	**g**	**m**	**g**
Ecoli	0.00001	0.001026842	0.002034	0.00110737	0.002628	0.000917608	0.002566
-	0.00004	0.001764815	0.002859	0.001975598	0.004129	0.001310417	0.003145
PAGI	0.00008	0.001681203	0.002236	0.002201538	0.004293	0.001917447	0.004506
	0.0001	0.00160251	0.001915	0.003472222	0.00715	0.002956853	0.007223
9a5c	0.00001	0.000159945	0.000148	0.000413169	0.000825	0.000551452	0.001258
-	0.00004	0.000424324	0.000314	0.000927381	0.001558	0.001297129	0.002711
PAGI	0.00008	0.00100396	0.001014	0.0016	0.00272	0.002632536	0.005502
	0.0001	0.001122227	0.001133	0.001997252	0.003489	0.002608159	0.005492
Subtilis	0.00001	0.001263787	0.003804	0.001378711	0.004922	0.001155789	0.004392
-	0.00004	0.001012736	0.002147	0.002205455	0.007278	0.001911494	0.006652
PAGI	0.00008	0.001048667	0.001573	0.002548377	0.007849	0.002321515	0.007661
	0.0001	0.001010093	0.001271	0.002816428	0.008538	0.002646329	0.00864
Aeruginosa	0.00001	0.000347097	0.000346	0.000515507	0.000795	0.000714339	0.001428
-	0.00004	0.000727451	0.000371	0.000969027	0.001095	0.001281921	0.002269
PAGI	0.00008	0.001036538	0.000539	0.001288571	0.001353	0.001679651	0.002889
	0.0001	0.001192904	0.000585	0.00136476	0.001436	0.002119626	0.003604
Griseus	0.00001	0.000696129	0.001446	0.000734002	0.001717	0.000807158	0.002385
-	0.00004	0.000948175	0.001299	0.000918519	0.001488	0.000915837	0.002024
PAGI	0.00008	0.001449153	0.00171	0.001389542	0.002126	0.001296759	0.002801
	0.0001	0.001599199	0.001837	0.001674015	0.002506	0.001457906	0.003053
Ecoli	0.00001	0.000776085	0.001202	0.000783963	0.001707	0.000769204	0.002172
-	0.00004	0.001455773	0.001315	0.001005546	0.001378	0.000952272	0.002085
BSGI	0.00008	0.002756193	0.002192	0.001872621	0.002361	0.00187713	0.003966
	0.0001	0.003293114	0.002702	0.002353124	0.003032	0.002016665	0.004187
9a5c	0.00001	0.000793818	0.002111	0.000781195	0.00221	0.000826046	0.00256
-	0.00004	0.00102073	0.001487	0.001012921	0.001803	0.001203159	0.002864
BSGI	0.00008	0.001561995	0.001838	0.001653401	0.002659	0.002381293	0.005326
	0.0001	0.001924911	0.00222	0.002204165	0.003472	0.002675843	0.005832
Subtilis	0.00001	0.002406199	0.00354	0.001313106	0.002653	0.001140405	0.00279
-	0.00004	0.005535173	0.007302	0.002238163	0.003947	0.001525763	0.003355
BSGI	0.00008	0.004352866	0.005445	0.003222746	0.005469	0.002196179	0.004713
	0.0001	0.003090641	0.003778	0.004004695	0.006995	0.003442809	0.0076
Aeruginosa	0.00001	0.000706657	0.002243	0.000713649	0.002403	0.000799112	0.002808
-	0.00004	0.001062861	0.001985	0.00102183	0.002186	0.001401433	0.003814
BSGI	0.00008	0.001402204	0.002074	0.00160984	0.003102	0.002371602	0.00616
	0.0001	0.001915596	0.002787	0.001901994	0.003501	0.002677713	0.006739
Griseus	0.00001	0.000819463	0.002885	0.000814174	0.003226	0.000691756	0.002884
-	0.00004	0.001215942	0.002758	0.001108942	0.002897	0.001133506	0.003408
BSGI	0.00008	0.001643389	0.003024	0.001920728	0.004434	0.002043195	0.005856
	0.0001	0.0017563	0.003011	0.002188104	0.004981	0.002365471	0.006774
Ecoli	0.00001	0.000411539	0.000408	0.000462691	0.000599	0.00057094	0.000969

- ECGI	0.00004	0.00088627	0.000519	0.000988718	0.000964	0.001203063	0.001791
	0.00008	0.001211374	0.000639	0.001114535	0.001049	0.001436026	0.002064
	0.0001	0.001392143	0.000808	0.00123621	0.001143	0.001871695	0.00269
9a5c - ECGI	0.00001	0.000344612	0.000607	0.000490983	0.00104	0.000552343	0.001299
	0.00004	0.000612934	0.000508	0.000775343	0.001044	0.000895195	0.001634
	0.00008	0.000813492	0.000533	0.000994662	0.001267	0.001124544	0.001972
	0.0001	0.000973532	0.000743	0.001244454	0.001683	0.001423138	0.002582
Subtilis - ECGI	0.00001	0.000498192	0.000937	0.000907299	0.001933	0.000731941	0.001836
	0.00004	0.001089876	0.001842	0.00143898	0.002844	0.001322625	0.002989
	0.00008	0.001262782	0.00205	0.00222532	0.00431	0.001866726	0.004216
	0.0001	0.001134042	0.001643	0.002560522	0.004823	0.002306108	0.00504
Aeruginosa - ECGI	0.00001	0.000504911	0.001825	0.000556753	0.001963	0.000595248	0.002182
	0.00004	0.000652377	0.001397	0.000747801	0.001734	0.000777888	0.002108
	0.00008	0.0008015	0.001282	0.001005982	0.002035	0.00111206	0.002734
	0.0001	0.000974784	0.001469	0.001240126	0.002449	0.001287705	0.003142
Griseus - ECGI	0.00001	0.000612225	0.002462	0.00055478	0.002265	0.00055153	0.002405
	0.00004	0.00081185	0.002195	0.000820896	0.002442	0.000796819	0.002675
	0.00008	0.001023211	0.002354	0.001056831	0.002821	0.001247257	0.003979
	0.0001	0.001169973	0.002595	0.001424164	0.003831	0.001400519	0.004372
Ecoli - SCGI	0.00001	0.001434189	0.003866	0.004017142	0.015795	0.00161837	0.00678
	0.00004	0.001611431	0.003107	0.005829785	0.020673	0.002219329	0.008097
	0.00008	0.001804491	0.002732	0.00470301	0.013967	0.002295806	0.007117
	0.0001	0.001445747	0.001948	0.004844245	0.01482	0.002430858	0.007638
9a5c - SCGI	0.00001	0.001305809	0.002758	0.001641339	0.004385	0.001373618	0.004285
	0.00004	0.003267109	0.005953	0.003219562	0.007584	0.001725225	0.004542
	0.00008	0.00347529	0.005239	0.003054996	0.006216	0.002593665	0.006272
	0.0001	0.003800621	0.005387	0.003553715	0.007165	0.002783452	0.006459
Subtilis - SCGI	0.00001	0.000845369	0.002394	0.005188847	0.025718	0.004160507	0.021333
	0.00004	0.001018693	0.001891	0.006014738	0.025793	0.00531558	0.023514
	0.00008	0.001121212	0.001591	0.006970816	0.027946	0.00688381	0.028764
	0.0001	0.001091804	0.00142	0.00587597	0.02166	0.005082207	0.019474
Aeruginosa - SCGI	0.00001	0.001606986	0.002199	0.001329329	0.002501	0.001882804	0.004373
	0.00004	0.002148197	0.00227	0.001939501	0.003084	0.003384353	0.007341
	0.00008	0.002336182	0.002132	0.002920371	0.004665	0.003820562	0.008521
	0.0001	0.002998412	0.002644	0.002957222	0.004369	0.003210351	0.006848
Griseus - SCGI	0.00001	0.000603656	0.0007	0.000815391	0.001121	0.001316295	0.002328
	0.00004	0.001230469	0.00126	0.001585076	0.002137	0.004529634	0.008147
	0.00008	0.001854088	0.001873	0.003258115	0.004346	5.30594E-05	0.000094
	0.0001	0.003411514	0.00384	0.005952196	0.008442	3.60992E-05	0.000069

PAGI: P. *aeruginosa* Genomic Island, BSGI: B. *subtilis* Genomic Island, ECGI: E.*coli* Genomic Island, SCGI: S.*coelicolor* Genomic Island, Ecoli: E.*coli* K12, 9a5c: X. *Fastidiosa* 9a5c strain, Subtilis: B.*subtilis* sub 168, Aeruginosa: P. *aeruginosa*, Griseus: S.*griseus*, m and g are both parameters of the Verhulst Equation where m = g/k (see methods). E. *coli* K12 genomic island and S. *coelicolor* genomic island in combination with five targets are not displayed in this table.

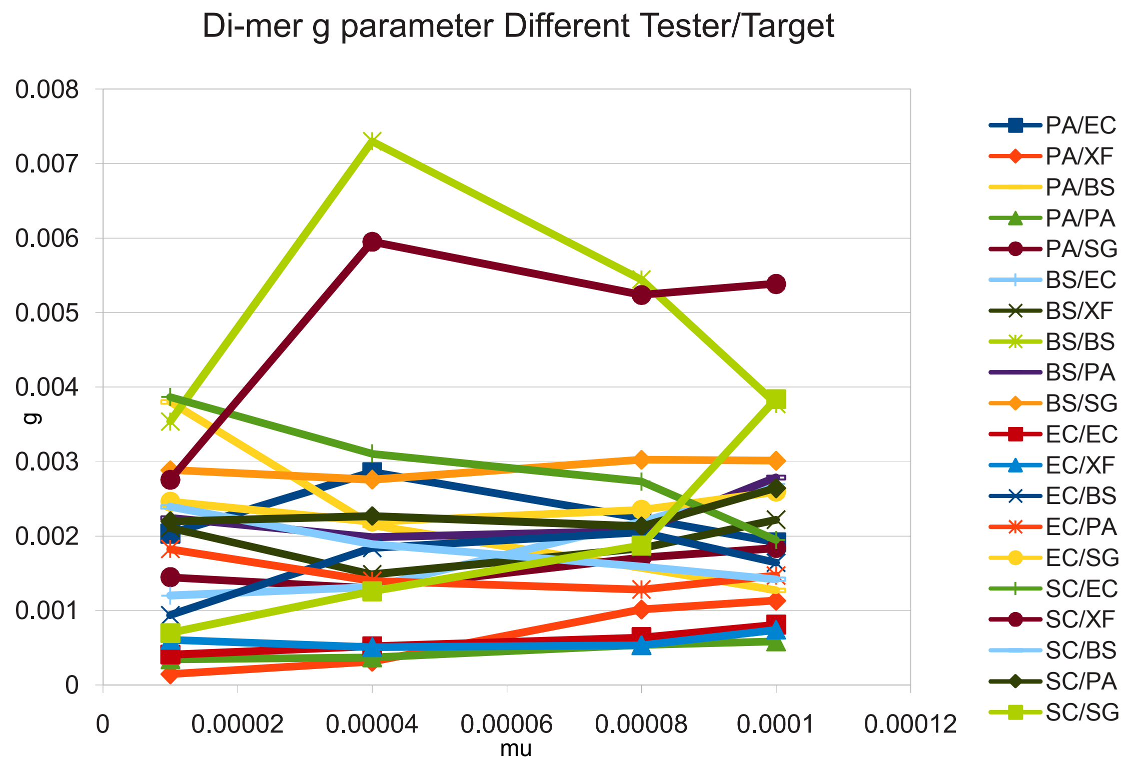


Fig. 2. Graph plot of di-mer g parameter estimate of the Verhulst Equation of all 20 combinations of tester and target. From the graph, no clear relationship can be stated between g and the other factors (μ, tester and target internal variance). Some combinations that show significant difference to the rest are BS/BS (B.*subtilis*), PA/SG (P.*aeruginosa*/S.*griseus*) and SC/EC (S.*coelicolor*/E.*coli*). This could be caused by extreme K values as stated in the results section.

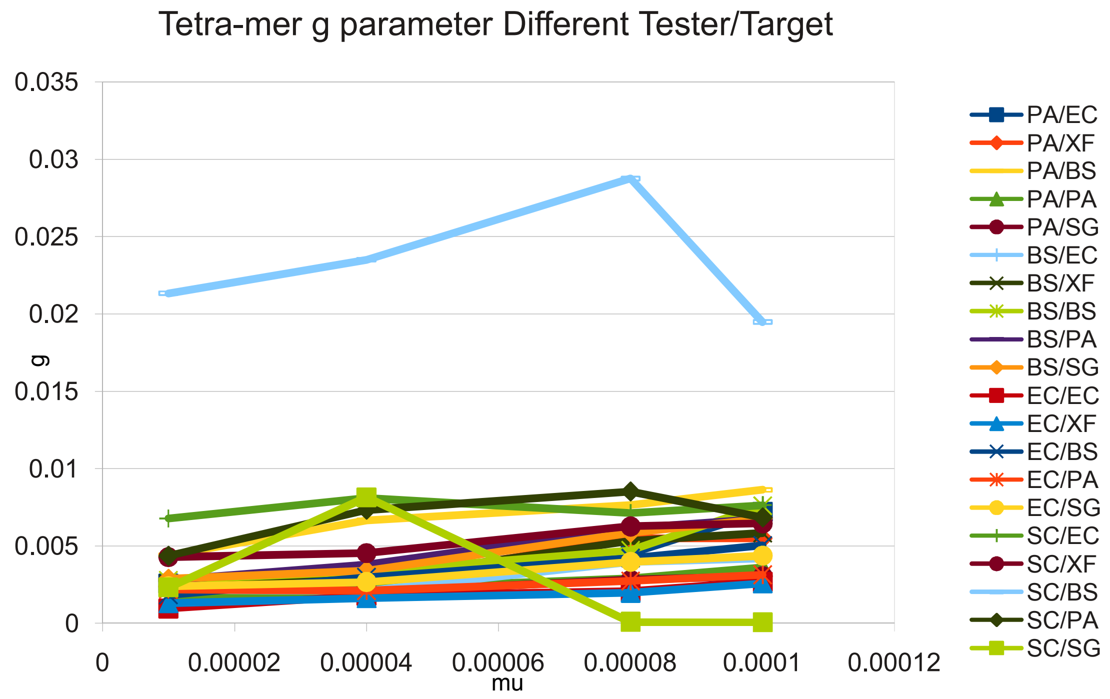


Fig. 3. Graph plot of tetra-mer g parameter estimate of the Verhulst Equation of all 20 combinations of tester and target. There are clear differences to figure 2 and similarities to tri-mer g plot in figure 15 result section. There is a clear trend of linearity between g and the three factors μ, tester and target internal variance with some outlier combinations such as SC/BS (S.*coelicolor*/B.*subtilis*) and (S.*coelicolor*/ S.*griseus*).

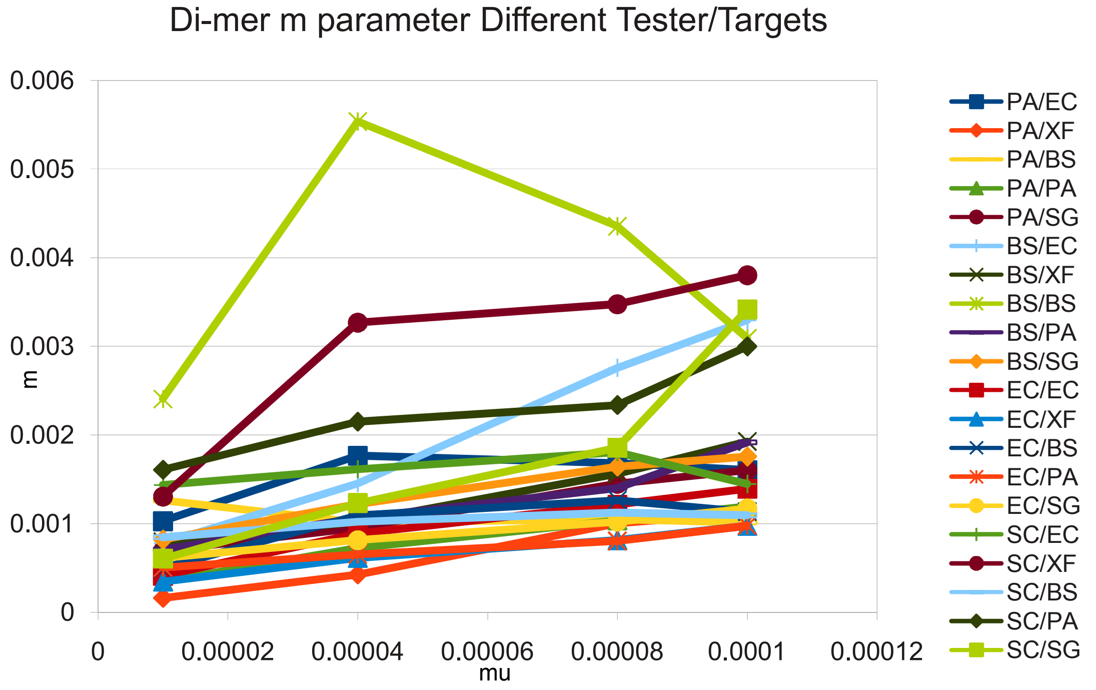


Fig. 4. Graph plot of di-mer m parameter estimate of the Verhulst Equation of all 20 combinations of tester and target. Similar to the graph of parameter 2g in figure 2 with the same outliers but there is a clear trend of a linear function existence. Outliers include BS/BS (B.*subtilis*), PA/SG (P.*aeruginosa*/S.*griseus*).

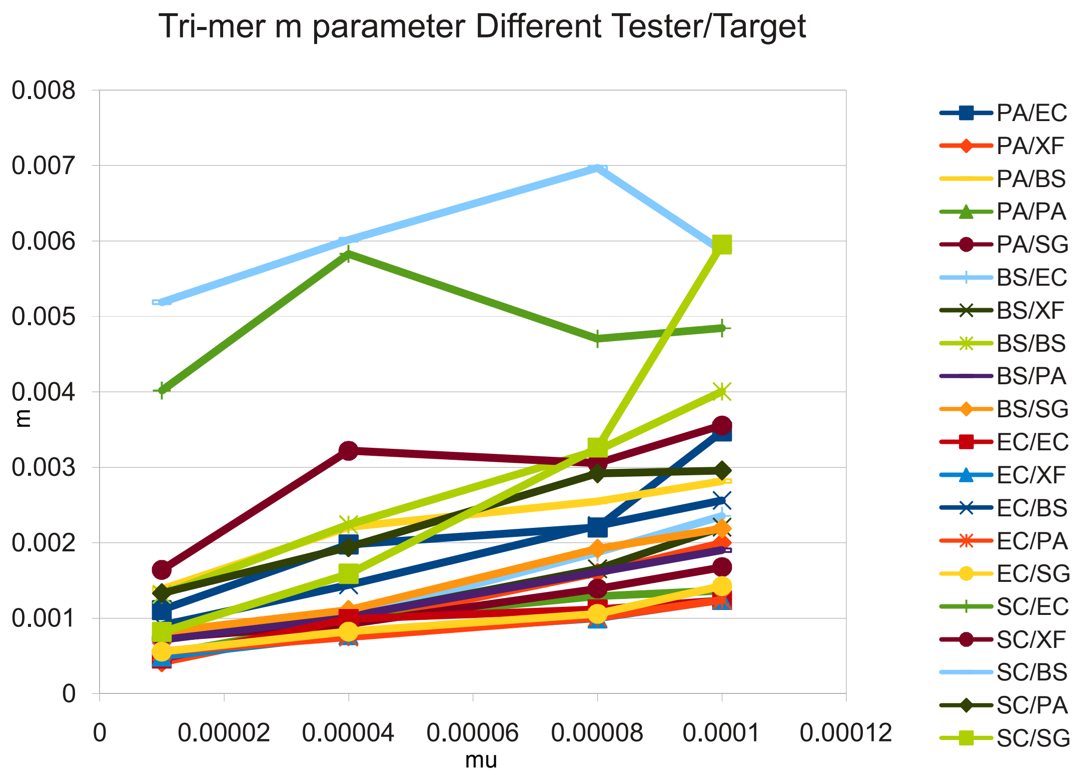


Fig. 5. Graph plot of tri-mer m parameter estimate of the Verhulst Equation of all 20 combinations of tester and target. Similar to the graph of parameter 3g in figure 15 in result section with the same outliers and a definite linear function existence. Outliers include SC/BS (S.*coelicolor*/B.*subtilis*) and (S.*coelicolor*/ E.*coli*).

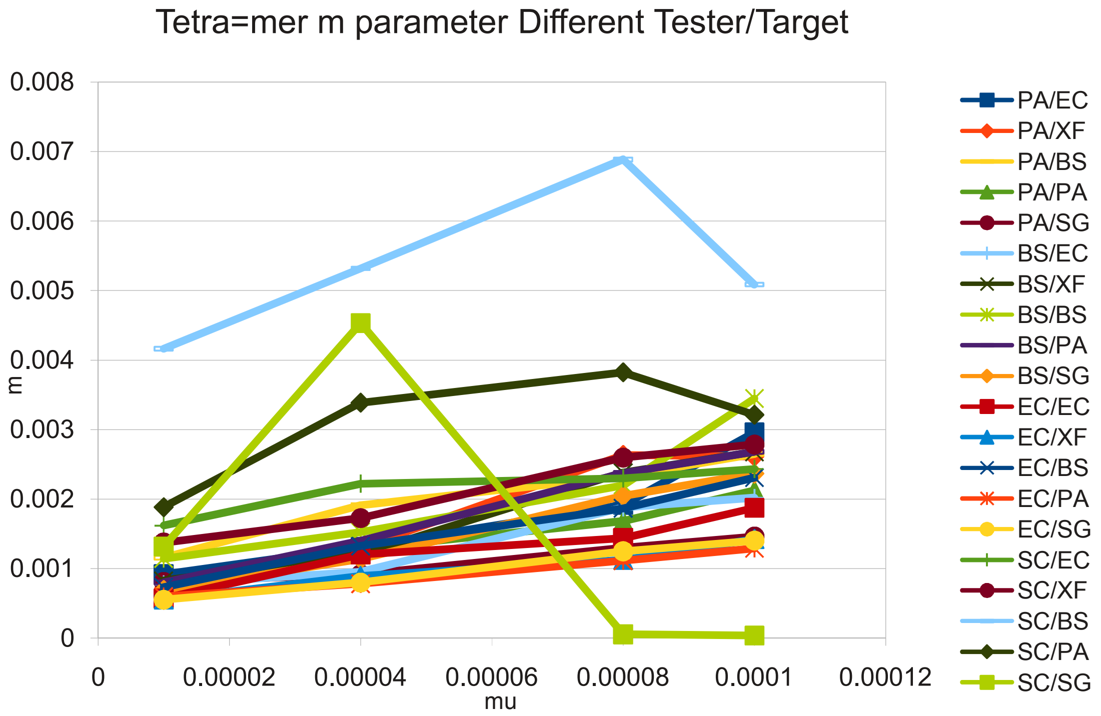
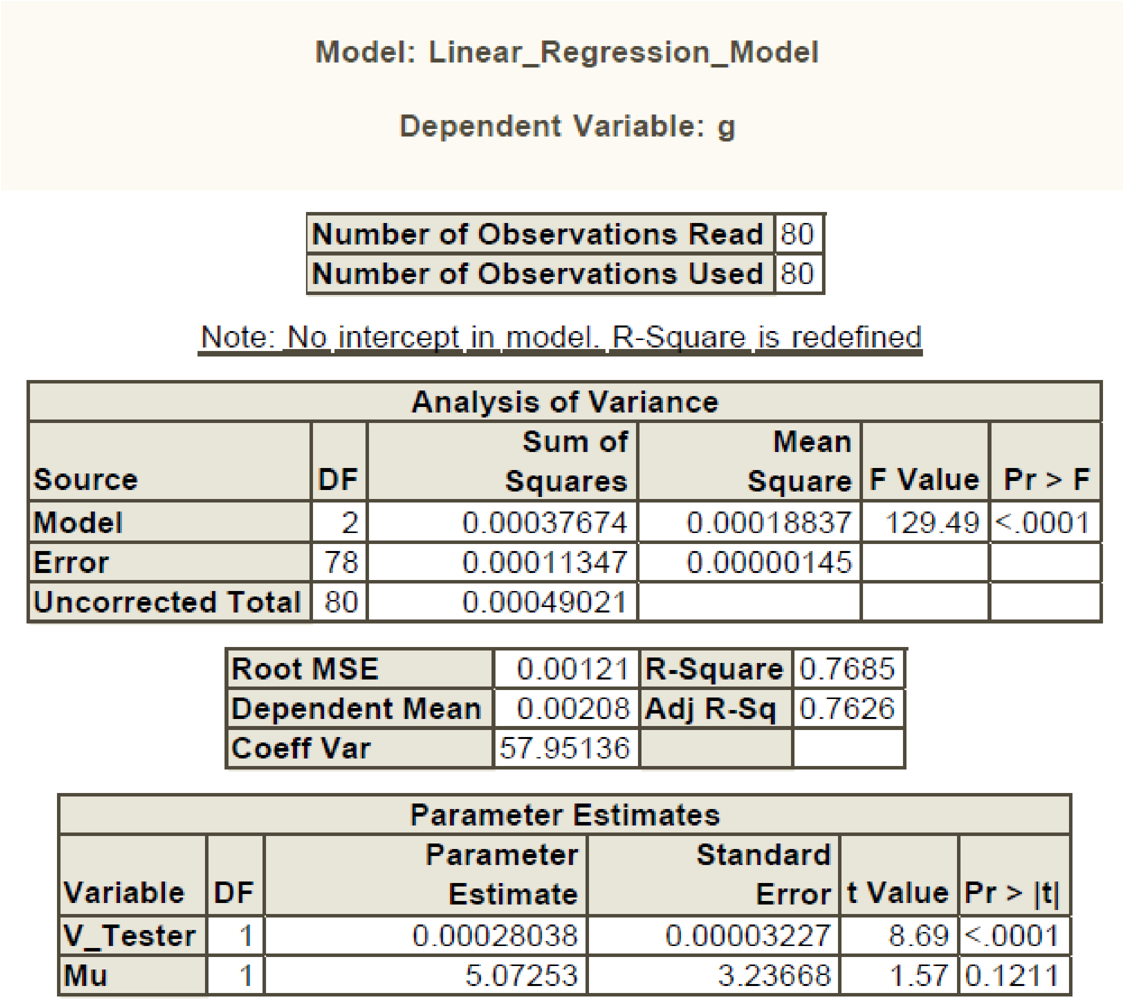


Fig. 6. Graph plot of tetra-mer m parameter estimate of the Verhulst Equation of all 20 combinations of tester and target. Exact same outliers as parameter 4g and has the same trend as parameter 2m and 3m. Since m parameter is calculated as g/K, the outliers' existence is the same since the parameters are in proportion to each other. Outliers include SC/BS (S.*coelicolor*/B.*subtilis*) and (S.*coelicolor*/ S.*griseus*).



Model: Linear_Regression_Model

Dependent Variable: g

Number of Observations Read	80
Number of Observations Used	80

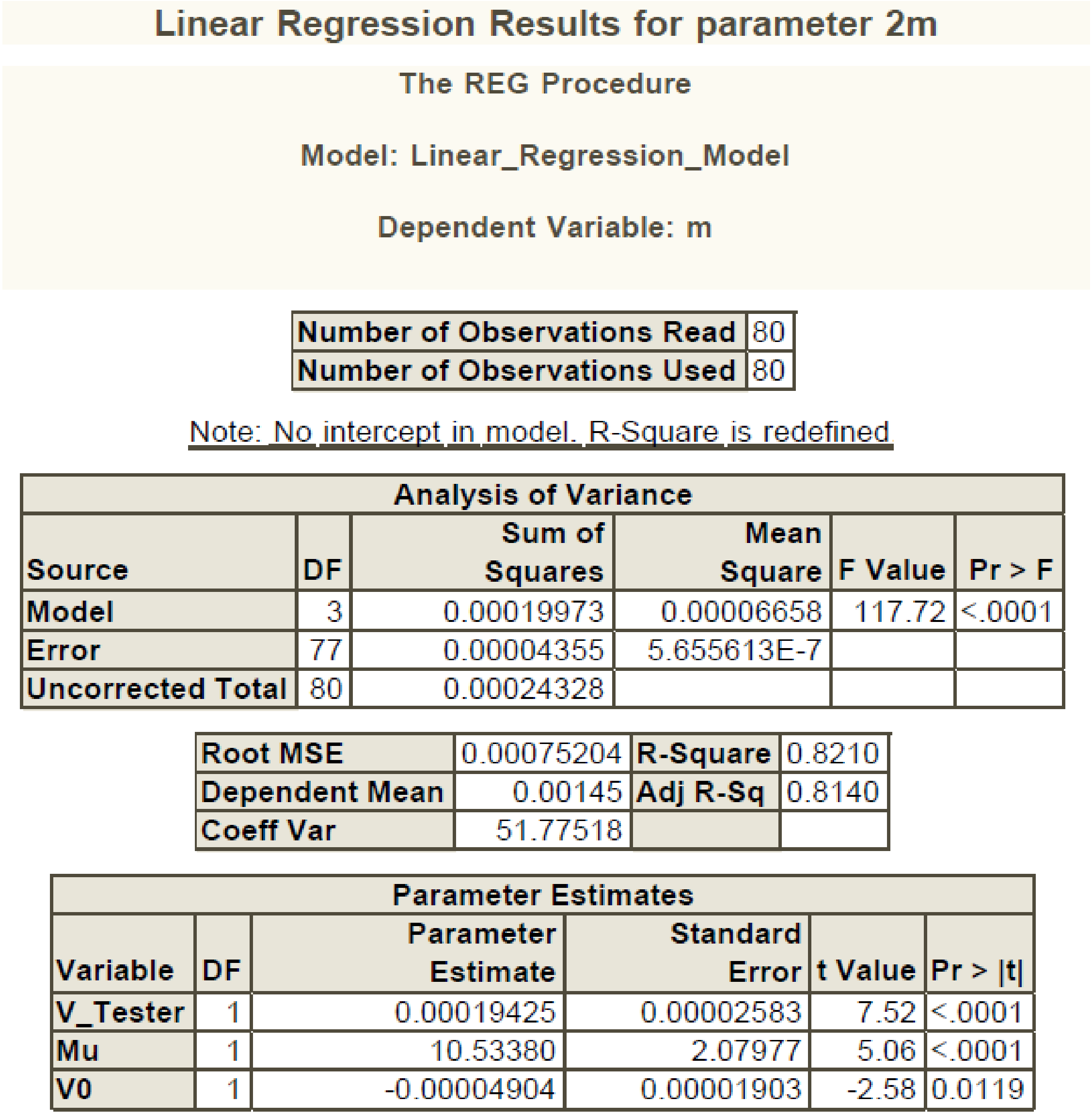
Note: No intercept in model. R-Square is redefined

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	0.00037674	0.00018837	129.49	<.0001
Error	78	0.00011347	0.00000145		
Uncorrected Total	80	0.00049021			

Root MSE	0.00121	R-Square	0.7685
Dependent Mean	0.00208	Adj R-Sq	0.7626
Coeff Var	57.95136		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
V_Tester	1	0.00028038	0.00003227	8.69	<.0001
Mu	1	5.07253	3.23668	1.57	0.1211

Fig. 7. Multi-variate regression analysis with all combinations of tester, target and µ for parameter 2g. The adjusted R-squared value is 76.26% which shows that there is a linear relationship but the model is poor in estimating parameter g. The significant variables are V_Tester and µ where µ is only significant at a 15% confidence level.



Linear Regression Results for parameter 2m

The REG Procedure

Model: Linear_Regression_Model

Dependent Variable: m

Number of Observations Read	80
Number of Observations Used	80

Note: No intercept in model. R-Square is redefined

Analysis of Variance

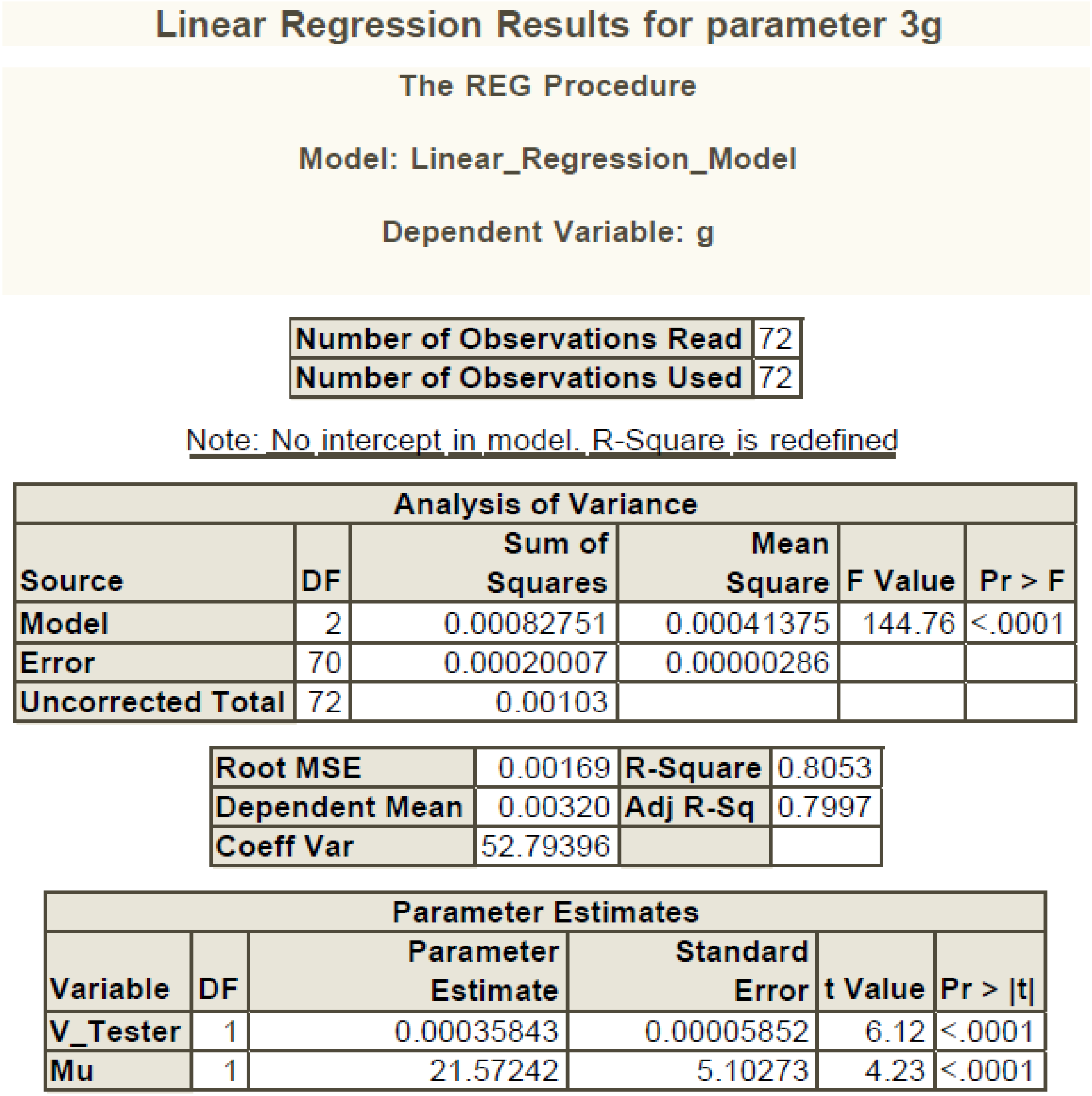
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	0.00019973	0.00006658	117.72	<.0001
Error	77	0.00004355	5.655613E-7		
Uncorrected Total	80	0.00024328			

Root MSE	0.00075204	R-Square	0.8210
Dependent Mean	0.00145	Adj R-Sq	0.8140
Coeff Var	51.77518		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
V_Tester	1	0.00019425	0.00002583	7.52	<.0001
Mu	1	10.53380	2.07977	5.06	<.0001
V0	1	-0.00004904	0.00001903	-2.58	0.0119

Fig. 8. Multi-variate regression analysis with all combinations of tester, target and μ for parameter 2m. The adjusted R-squared value is 81.40% which is a better model than 2g in terms of estimating 2m with variables V_Tester, Mu and V0 where all three variables are significant at 5% confidence level.



Linear Regression Results for parameter 3g

The REG Procedure

Model: Linear_Regression_Model

Dependent Variable: g

Number of Observations Read	72
Number of Observations Used	72

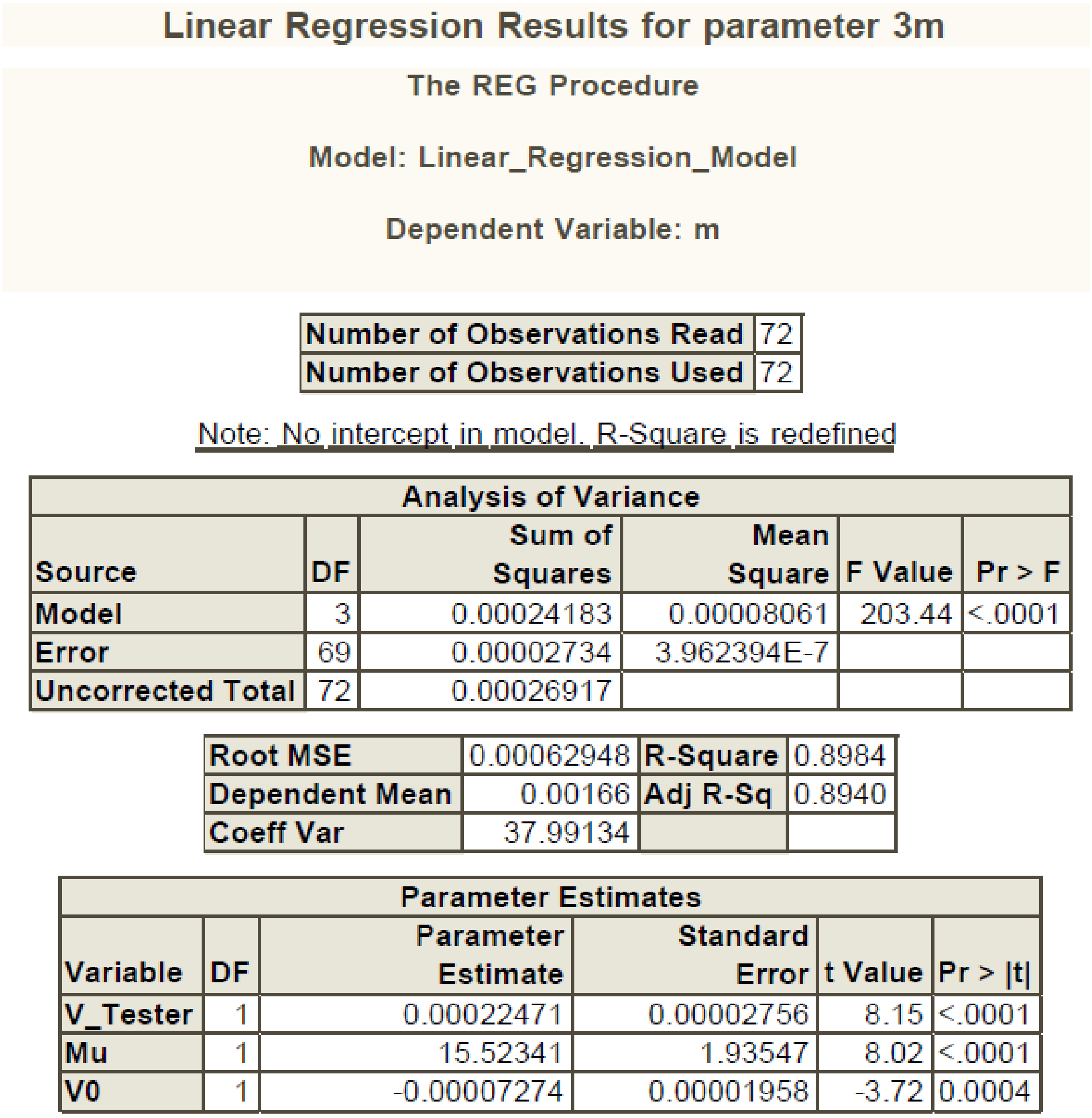
Note: No intercept in model. R-Square is redefined

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	0.00082751	0.00041375	144.76	<.0001
Error	70	0.00020007	0.00000286		
Uncorrected Total	72	0.00103			

Root MSE	0.00169	R-Square	0.8053
Dependent Mean	0.00320	Adj R-Sq	0.7997
Coeff Var	52.79396		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
V_Tester	1	0.00035843	0.00005852	6.12	<.0001
Mu	1	21.57242	5.10273	4.23	<.0001

Fig. 9. Multi-variate regression analysis with all combinations of tester, target and µ for parameter 3g. The adjusted R-squared value is 79.97% which is a better model than parameter 2g but still a poor model where 20% of the data will be estimated with error. The variables are the same as parameter 2g including V_Tester and Mu where both variables are significant at 1% confidence level. In combination with results of parameter 4g in figure 15 results section, parameter g is a function of V_Tester, V_Target and Mu.



Linear Regression Results for parameter 3m

The REG Procedure

Model: Linear_Regression_Model

Dependent Variable: m

Number of Observations Read	72
Number of Observations Used	72

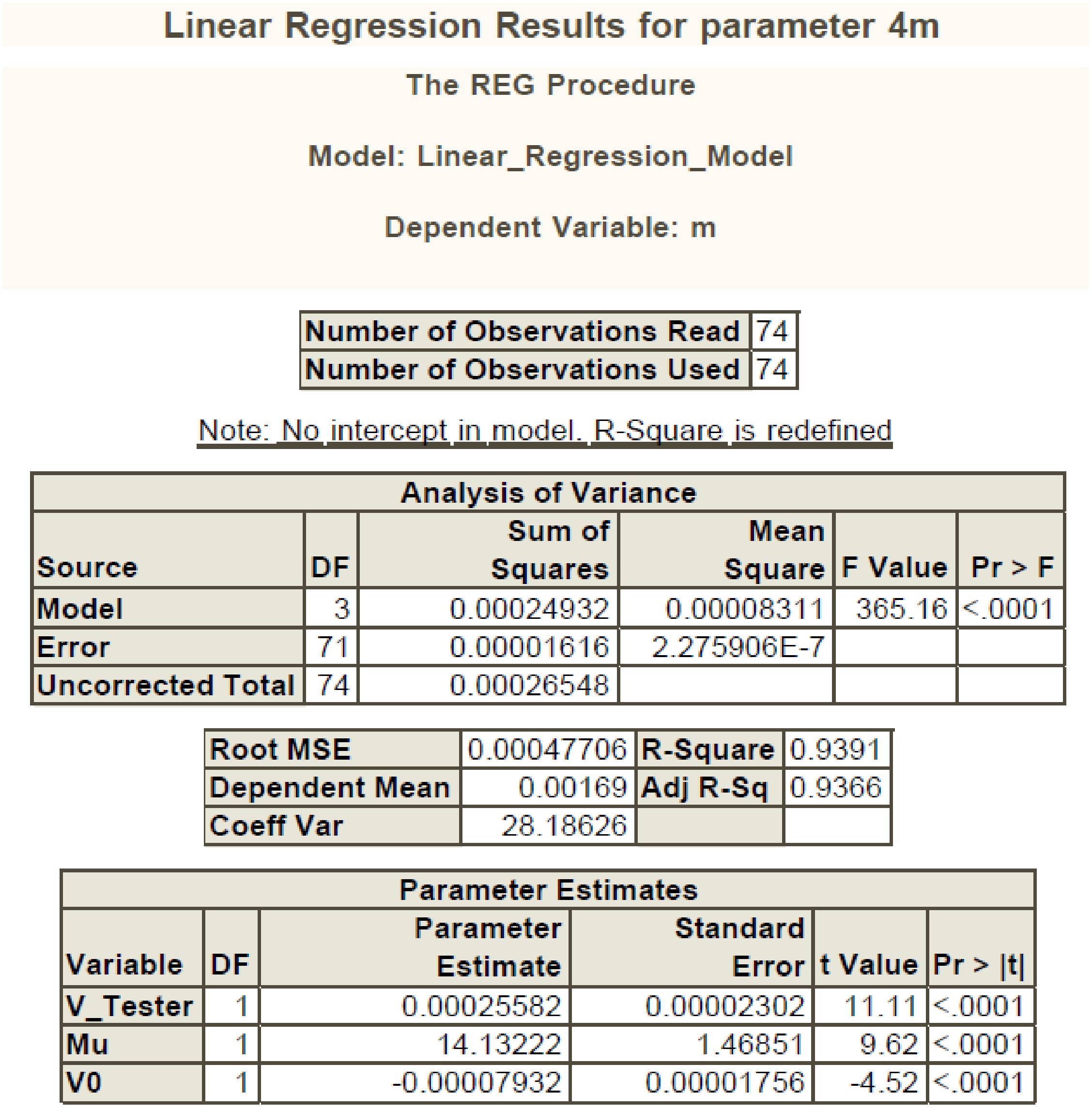
Note: No intercept in model. R-Square is redefined

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	0.00024183	0.00008061	203.44	<.0001
Error	69	0.00002734	3.962394E-7		
Uncorrected Total	72	0.00026917			

Root MSE	0.00062948	R-Square	0.8984
Dependent Mean	0.00166	Adj R-Sq	0.8940
Coeff Var	37.99134		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
V_Tester	1	0.00022471	0.00002756	8.15	<.0001
Mu	1	15.52341	1.93547	8.02	<.0001
V0	1	-0.00007274	0.00001958	-3.72	0.0004

Fig. 10. Multi-variate regression analysis with all combinations of tester, target and µ for parameter 3m. The adjusted R-squared value is 89.40% which is a very good model in terms of estimating parameter 3m. The three significant variables are V_Tester, Mu and V0 where all three are significant at 1% confidence level.



Linear Regression Results for parameter 4m

The REG Procedure

Model: Linear_Regression_Model

Dependent Variable: m

Number of Observations Read	74
Number of Observations Used	74

Note: No intercept in model. R-Square is redefined

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	0.00024932	0.00008311	365.16	<.0001
Error	71	0.00001616	2.275906E-7		
Uncorrected Total	74	0.00026548			

Root MSE	0.00047706	R-Square	0.9391
Dependent Mean	0.00169	Adj R-Sq	0.9366
Coeff Var	28.18626		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
V_Tester	1	0.00025582	0.00002302	11.11	<.0001
Mu	1	14.13222	1.46851	9.62	<.0001
V0	1	-0.00007932	0.00001756	-4.52	<.0001

Fig. 11. Multi-variate regression analysis with all combinations of tester, target and μ for parameter 4m. The adjusted R-squared value is 93.66% which is the best model out of all six parameters. Based on the R-squared value, we can assume that parameter 4m follows a multi-variate linear function. The significant variables are V_Tester, Mu and V0 which are all significant at 1% confidence level. Parameter m for all K-mers also follows the same function of V_Tester, Mu and V0.

Table 2. Differential Estimate for different simulated sequences

Internal 4V	Distance	V_0	FMR	Differential
5.17	7.67	2.97	1.483559	0.884918069
5.28	7.2	2.8	1.363636	0.833634975
5.27	6.97	2.77	1.322581	0.841290623
5.35	6.83	2.68	1.276636	0.817837767
5.38	6.44	2.59	1.197026	0.810451008
5.41	6.66	2.6	1.231054	0.792621145
5.42	6.49	2.57	1.197417	0.786001884
5.47	6.48	2.52	1.184644	0.768645337
5.53	6.35	2.46	1.148282	0.751016954
5.53	6.24	2.45	1.128391	0.7442572
5.55	6.21	2.41	1.118919	0.731611437
5.58	5.87	2.34	1.051971	0.726944162
5.59	6.13	2.37	1.096601	0.717473583
5.61	6.18	2.36	1.101604	0.707036492
5.63	6.12	2.33	1.087034	0.696544331
5.67	6.15	2.3	1.084656	0.683130736
5.68	5.99	2.27	1.054577	0.67405555
5.71	6.03	2.26	1.056042	0.664996225
5.72	5.59	2.19	0.977273	0.663905307
5.74	5.83	2.21	1.015679	0.65813006
5.75	6.02	2.22	1.046957	0.649327844
5.77	5.92	2.19	1.025997	0.641590694
5.79	5.73	2.16	0.989637	0.637581008
5.79	5.8	2.16	1.001727	0.632360139
5.82	5.81	2.14	0.998282	0.626294694
5.82	5.47	2.09	0.939863	0.624855445
5.83	5.53	2.1	0.948542	0.623340175
5.85	5.74	2.12	0.981197	0.619581157
5.86	5.66	2.1	0.96587	0.616483633
5.87	5.44	2.06	0.926746	0.615368196
5.88	5.53	2.08	0.940476	0.614299524
5.9	5.32	2.02	0.901695	0.613437464
5.9	5.52	2.06	0.935593	0.611780512
5.9	5.42	2.04	0.918644	0.610724923
5.92	5.48	2.05	0.925676	0.609730404
5.92	5.4	2.02	0.912162	0.608275093
5.93	5.35	1.99	0.902192	0.606008331
5.94	5.42	1.99	0.912458	0.602999982
5.97	5.36	1.96	0.897822	0.599743379
5.96	5.27	1.95	0.884228	0.597514959
5.99	5.36	1.95	0.894825	0.594559652
6	5.38	1.94	0.896667	0.591144048
6.01	5.22	1.9	0.868552	0.58816412
6.04	5.45	1.92	0.902318	0.583786152
6.04	5.45	1.92	0.902318	0.579836481
6.05	5.28	1.87	0.872727	0.575811842

6.06	5.5	1.91	0.907591	0.571681541
6.08	5.59	1.9	0.919408	0.566520698
6.09	5.52	1.89	0.906404	0.562199573
6.1	5.31	1.86	0.870492	0.559338375
6.14	5.31	1.84	0.864821	0.556181928
6.16	5.37	1.82	0.871753	0.551806326
6.2	5.48	1.82	0.883871	0.546960748
6.22	5.69	1.82	0.914791	0.540414614
6.24	5.82	1.82	0.932692	0.533239391
6.25	5.9	1.84	0.944	0.526948598
6.26	6	1.86	0.958466	0.521234098
6.28	6.02	1.86	0.958599	0.5158925
6.29	6.21	1.89	0.987281	0.510756658
6.29	6.34	1.9	1.007949	0.505413194

Internal 4V: Tetra-mer internal variance of tester, Distance: Absolute distance measure between tester and target (See section 2.2 in Literature Review), V_0: Variation between composition pattern between tester and target, FMR: Forward Mutation Ratio, Differential: Empirical differential calculated based on FMR and V_0, also equal to the mutation rate.