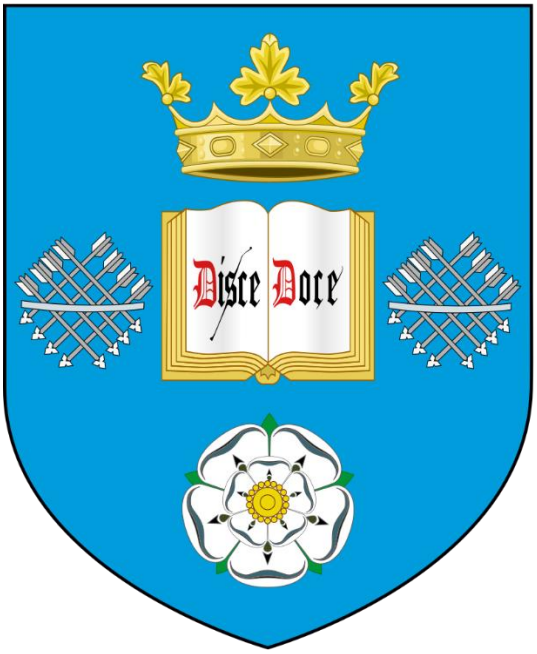
The development and application of a random mutagenic technique for the mutation and directed evolution of Reverse Transcriptase

Submitted to the University of Sheffield for the degree of Doctor of Philosophy

By

James Charles Florence

School of Biosciences



December, 2022

Abstract

Directed evolution is a powerful tool in the arsenal of the molecular biologist. It provides scientists with the ability to harness the power of evolution. The significance of the discovery of directed evolution was recognized by the award of the 2018 Nobel prize in 2018 to Frances Arnold.

Reverse transcriptase is an enzyme with many different functions, foremost among them being RNA dependent DNA polymerisation. This function has made reverse transcriptase a key enzyme for both molecular biology research and diagnostic applications. There is a demand for reverse transcriptase with increased functionality, be it increased thermotolerance enabling the enzyme to retain activity at temperatures above the denaturation point of RNA, or increased terminal transferase activity, to better facilitate template switching *in vitro*. Reverse transcriptase is therefore an ideal candidate for directed evolution.

In the course of this work, a novel statistical model for the introduction of mutations over the course of an error prone polymerase chain reaction is described. This model is then validated by the mutagenesis and subsequent sequencing of three model systems: an engineered form of a bacteria rubredoxin protein, reverse transcriptase, and a polyethylene terephthalate dehydrogenase. The gene encoding reverse transcriptase is mutagenised and expressed in a recombinant form, the resultant mutants are assayed for increased RNA-dependent DNA-polymerase activity at 65℃. The work carried out in this thesis represents an attempt to better understand the process by which mutations are introduced via error-prone DNA polymerization. It is posited how these advancements can help to minimise the number of rounds of directed evolution that are required before an appropriate end point is reached.

Table of Contents

Chapter one

General Introduction

Chapter two

Materials and Methods

Chapter three
Development of a tuneable random mutagenic technique

Chapter four
Validation of the probabilistic model of EP-PCR

Chapter five
Functional assay and screening of random mutant libraries

Chapter six
General Discussion

Abbreviations

DNA – DeoxyriboNucleic Acid

RNA – RiboNucleic Acid

DdDp – DNA-dependent DNA polymerase

RdDp – RNA-dependent DNA polymerase

RTase – Reverse Transcriptase

NiFe – Iron/Nickel binding protein (p53-rubredoxin fusion)

PETase – PolyEthyl Terephthalate hydrolase

PfuEP - Error-prone recombinant polymerase from *Pyrococcus furiosus*

PhoEP – Error-prone recombinant polymerase from *Pyrococcus horikoshii*

PCR – Polymerase Chain Reaction

RT-PCR – Reverse Transcription Polymerase Chain Reaction

qPCR – quantitative Polymerase Chain Reaction

RT-qPCR – Reverse Transcription quantitative Polymerase Chain Reaction

rNTP – riboNucleotide TriPhosphate

dNTP – deoxyNucleotide TriPhosphate

G – Guanine

A – Adenine

T – Thymine

C – Cytosine

Acknowledgements

I would like to thank Diagenode LLC for part sponsoring this project, as well as the University of Sheffield for awarding the Ken and Noreen Murray Biotech Scholarship to me.

Many thanks are due to Professor Simon Foster, and his lab team for providing a bench and a lab for me to carry out my work, as well as a plethora of advice and support.

Thanks are due to Alex Wakeman, who carried out some EP-PCR and sequencing experiments; as well as to Professor Mark Dickman who performed MS-MS analysis on my samples.

Finally, I would like to thank my supervisor, Professor David Hornby for providing an abundance of wisdom and knowledge throughout the course of my PhD.

1. Introduction

1.1 Directed evolution

Since the times of the lower Egyptian empire (about 5000 years ago) humans have been manipulating the process of evolution to their benefit; strains of grass were crossed either accidentally or purposely, to generate offspring that contained increased grains that could be more readily consumed by humans and animals. Since these humble beginnings, humans have continued to apply selective pressure on crops and animals - from increased muscle content on cows and bovine breeding stocks, to the domestication of wild dogs, humans have been manipulating genetics for millennia. However, it is only in the past century that humans have discovered the mechanism for these heritable changes to agriculture.

In 1866, Gregor Mendel, a monk at a monastery in Brno, published a paper, entitled "Experiments on Plant Hybridization", which related to the inheritance of phenotypes in pea plants. While this paper was mostly ignored in his lifetime, Mendel's work went on to lay the groundwork for the nascent field of genetics. Mendel's work showed that various features from a pea plant, such as flower colour, pea pod texture, and plant height, could be inherited by the offspring in a predictable and reproducible way. Mendel posited that these observations were due to heritable material in the plants, and that a dominant feature would always show instead of a recessive one (Bateson and Mendel, 2011). We know now that Mendel's heritable materials are genes, and are made of DNA, as discovered by Watson and Crick in 1953, and that Mendel's pea phenotypes were likely monogenic, due to the clear 3:1 ratios he found.

It wasn't until the latter half of the 20th century that the mechanism of Mendelian inheritance was discovered. There was widespread debate as to which component of the cell provides the heritable material that can be passed down to offspring. Many scientists believed that protein must be the elusive heritable material - as DNA and RNA are far simpler molecules, containing only 4 variant monomers as opposed to over 20 in processed polypeptides. DNA was proven to carry the genetic information in 1944, by Avery et al., a fact that was later cemented by Hershey and Chase (1952). Avery worked with R and S strain *Streptococcus pneumoniae*, the former being harmless, and the latter causing disease, and

ultimately death of mice. Avery based his experiments on work carried out by Griffith in 1928**,** wherein injecting mice with R strain *S. pneumoniae* did not kill the mice, while inoculation of the S strain did. If the *S. pneumoniae* had been heat killed prior to injection into the mice, inoculation of neither the R nor the S strain resulted in the death of the mice, however when heat killed S strain was inoculated alongside live R strain bacteria, the mice died. This was consistent with results reported by Griffiths; however, Avery went one step further to identify the cause of this transformation of *S. pneumoniae*. By performing various precipitations and washes, Avery was able to purify the fraction of the heat killed cells that caused transformation of the R strain to the S strain. By treating this fraction with various peptidases and RNase, Avery was able to show that the transformative material, and thus the material that contained the genetic material, was DNA (Avery et al., 1944). This conclusion was further verified 8 years later, when Hershey and Chase showed that inoculation of the T2 phage DNA into a host bacterium was sufficient to cause disease of the plant. By labelling either the sulphur in proteins, or the phosphorus in nucleotides with radioactivity, Hershey and Chase were able to show that only the DNA from T2 phage enters the cell, and therefore must contain the genetic information necessary to produce more T2 phage particles.

Once the biochemical to contain genetic information was known to be DNA, further questions into how exactly DNA carries and transfers this information from generation to generation. A big step in elucidating these facts came when the structure of DNA was characterised by James Watson and Francis Crick, based on work done by Rosalind Franklin, Maurice Wilkins, and Raymond Gosling. This now famous double helical structure was published in 1953, and contained the prescient line "...it has not escaped our notice that the specific pairing we have postulated immediately suggests a copying mechanism for the genetic material." (Watson and Crick, 1953). This copying mechanism alluded to by Watson and Crick would be clarified in 1958 by Meselson and Stahl, who showed that DNA replicated semi-conservatively (Meselson and Stahl, 1958). By using N^{15} in nucleotides, Meselson and Stahl were able to show that the increased density associated with N^{15} labelled nucleotides decreased generationally when grown in N^{14} containing media. The rate of this decrease in density provided evidence that DNA replicates semi conservatively.

Once the mechanism for inheritance was elucidated, scientists worked quickly to develop methods to manipulate individual genomes and genes. In the past 60 years, the first restriction endonuclease was identified and isolated, allowing manipulation of DNA on a molecular level (Smith and Welcox, 1970). This discovery paved the way for the manipulation of genetic information, with the first molecular cloning experiment being reported in 1972 (Jackson et al., 1972). The development of the polymerase chain reaction (PCR) allowed rapid and precise amplification of any gene or DNA fragment of interest (Saiki et al., 1985). This was quickly followed up with methods for reverse transcription PCR (RT-PCR) and quantitative PCR (qPCR), allowing quantification of the amount of DNA produced per cycle in a PCR (Higuchi et al., 1992).

It is thus only recently that humans have been able to isolate individual genes from any source. This advancement has allowed scientists to bring selective breeding and manmade selection into the modern era. Rather than relying on systems biology, and polygenic traits as in selective breeding, it is now possible to direct evolution to maximise the desired function of an individual gene. This work was pioneered by Frances Arnold, who used random mutagenesis and site directed mutagenesis to improve the activity of subtilisin E in high concentrations of polar organic solvents. It was found that four mutations could increase the peptide hydrolysing activity of subtilisin E by 256 fold in dimethylformamide (Chen and Arnold, 1993). This work marked the first time that the theory of evolution was harnessed by scientists and earned the Nobel Prize 25 years later.

The aim of this PhD is to attempt to combine the historic understanding of the biochemical properties of reverse transcriptase (RTase) with experimental, directed evolution in order to increase the thermostability and terminal transferase activity of this enzyme. To do this, mutagenesis and the subsequent screening and selection of beneficial mutants will be carried out. Additionally, this PhD will aim to elucidate a mathematical model for the introduction of mutations during error-prone polymerase chain reaction (EP-PCR), in order to maximise the efficiency of the directed evolution process.

1.2 Reverse transcriptase

In 1970, a Nature article heralded the discovery of RTase as the "Central Dogma Reversed" (Editorial Nature, 1970). This prompted a rebuttal from the grandfather of molecular genetics and author of the central dogma, Francis Crick, wherein he states that the central dogma does not disallow passage of information between nucleic acids, but rather than information cannot be passed from proteins back to nucleic acids (Crick, 1970). Nevertheless, in the same article in which he dismisses the "unsigned article" as misunderstanding the central dogma, Crick also highlights the "very important work" that first identified RTase, indicating the significance of the discovery.

The discovery of RTase was published simultaneously by two separate research groups (Baltimore, 1970; Temin and Mizutani, 1970) in which an RNA- dependent DNA polymerase was identified in Rauscher Mouse Leukaemia Virus (RMLV) and Rous Sarcoma Virus (RSV). Both groups extracted virus particles which were then treated with non-ionic detergents. The treated virus was then incubated with labelled ^{3}H-TTP in a polymerase assay, and the counts in the acid insoluble product were measured. This product was shown to be DNA as it could be made acid soluble by the addition of pancreatic deoxyribonuclease but was immune to the effects of pancreatic ribonuclease. Moreover, the template was shown to be RNA, as the addition of pancreatic ribonuclease to the reaction led to a decrease in the ^{3}H-TMP incorporated into the acid insoluble product.

Since its discovery, many features of RTase have been elucidated. RTase has been found to have many different activities, including: RNA-dependent DNA-polymerase (RdDp); DNA-dependent DNA-polymerase (DdDp); terminal transferase; and RNA endonuclease(Baltimore, 1970; Clark, 1988; Grandgenett et al., 1973; Temin and Mizutani, 1970). Additionally, it has been found that RTase can be either monomeric - Molony Murine Leukaemia Virus reverse transcriptase (MMLV RTase) - or dimeric - HIV-1 reverse transcriptase (HIV-1 RTase) (Das and Georgiadis, 2004; Jaeger et al., 1998).

This multifunctional nature of RTase could be indicative of its origins. Current theories postulate that RTase was present in a hybrid DNA/RNA genetic system that served as an intermediate between the earlier RNA-based life and the late DNA-based life (Leipe et al., 1999). This would mean that RTase is a very early protein, suggesting why it can perform multiple functions such as strand displacement, that normally would require several distinct proteins, although the distinct activities are of a relatively low catalytic efficiency compared with dedicated enzymes.

RTase has been regarded as a medically relevant protein since its discovery. Initially discovered in tumorigenic viruses, RTase has since been a prevalent target for HIV drugs – such as 3'-azido-3'-deoxythymidine (AZT)(Furman et al., 1986) – as it is critical in the life cycle of a retrovirus. By knocking out or down RTase activity, AZT prevents HIV from replicating upon entry to a new cell, however it does not cure HIV due to the incorporation of the HIV genome inside the host genome (Perez and Nolan, 2001).
In addition to its clinical relevance, RTase is also important in a laboratory setting. Techniques such as Reverse Transcription PCR (RT-PCR) and RNA-seq rely intrinsically on the ability of RTase to convert RNA to DNA (see Table 1). As such, it could be said that the field of transcriptomics is built on the back of RTase. The wild-type activity of RTase is not particularly conducive to laboratory practices; RTase is not thermostable meaning that all experiments must be carried out at a maximum of 42°C (Arezi and Hogrefe, 2009). This means that experiments can take a long time to complete, and that secondary structures within the RNA may have an effect on the outcome of the experiments. Additionally, RTase lacks a proofreading exonuclease domain, meaning that the fidelity of RTase is lower than that of other DNA polymerases (Menéndez-Arias, 2009). As such, there is some demand to find thermostable, high-fidelity mutants of RTase.

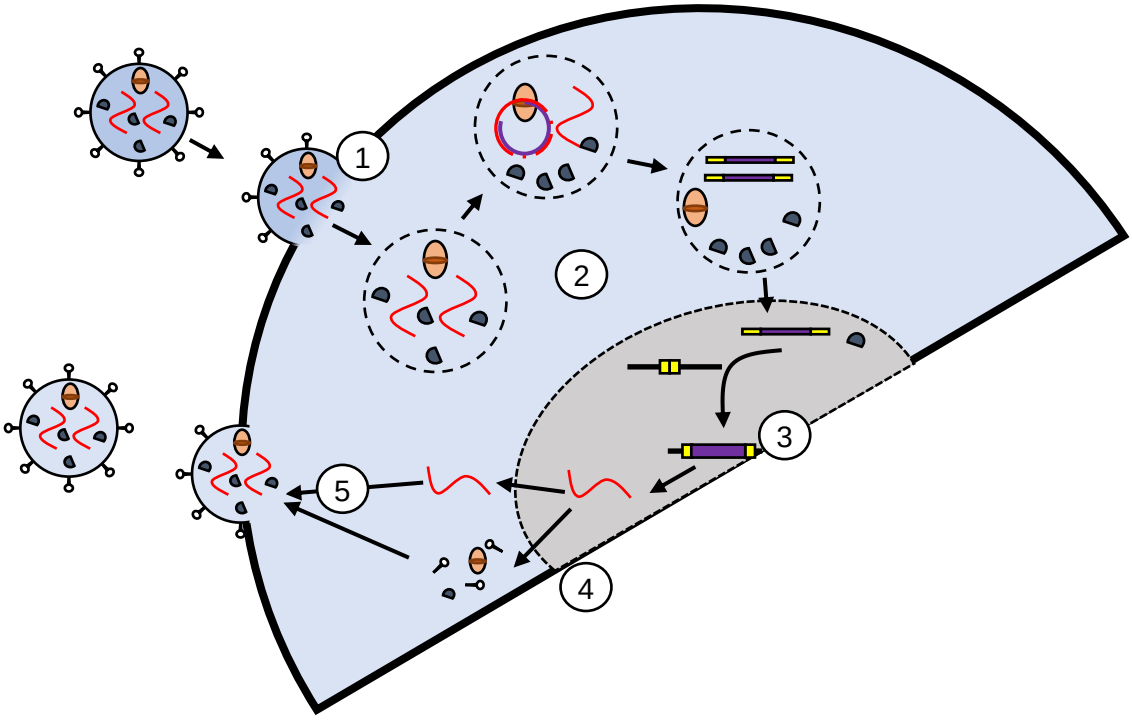


Figure 1: The lifecycle of a retrovirus. (1) The retrovirus binds to the host cell wall. In HIV-1 cells, this occurs via an interaction with CCR5 receptor. The viral contents are inserted into the host cell cytoplasm. (2) Reverse transcription of the retroviral genomic RNA occurs using reverse transcriptase. (3) The viral genome is integrated into the host genome through the action of an integrase protein. Long Terminal Repeats (LTRs) flanking the viral genome facilitate integration. (4) The integrated viral genome is transcribed into RNA by the host cell machinery. The transcribed RNA is then translated into a polyprotein which in turn is cleaved into viral capsid proteins and viral enzymes such as integrase and reverse transcriptase. (5) Viral capsid subunits auto-assemble into viral particles and viral genomic RNA and proteins are encapsulated by the capsid. The viral particle then buds off the host membrane (adapted from Perez and Nolan, 2001)

Table 1: Multiple different companies utilise reverse transcriptase in various products

Product	**C**apture and **A**mplification by **T**ailing and **S**witching (CATS)	NEBNext® Ultra™ RNA Library Prep Kit	LunaScript RT-qPCR
Company	Diagenode	NEB	NEB
Technique	RNA-Seq	RNA-Seq	RT-PCR
Time taken for reverse transcription	150 minutes	40 minutes	13 minutes
Strands synthesised by reverse transcriptase	2	1	1
Reverse Transcription temperature	42°C	42°C	55°C
Primers	Poly(T) and Template Switching Oligo (TSO)	Random	Poly(T) and Random hexamers

(Information taken from NEB.com, 2019A; NEB.com, 2019B; Diagenode.com, 2016)

1.2.1 Structure of RTase

The same domain architecture is present in all RTases; all RTases contain a ***fingers*** domain, a ***palm*** domain, a ***thumb*** domain, a ***connection*** domain and an ***RNase H*** domain. The former three domains are named due to their similarity to a right hand in the way that it grips the DNA:RNA duplex and is common to many other DNA polymerases such as DNA polymerase I (Kohlstaedt et al., 1992; Sawaya et al., 1994).

Despite an identical domain architecture, the structure of RTase has been found to vary; some RTases, including MMLV RTase, are monomeric while others, such as HIV-1 RTase, are dimeric. Regardless, it seems that both monomeric and dimeric RTases use the same mode

of action in order to polymerise DNA based on an RNA template. Highlighting this point, Painter et al., (1990, reviewed in Jacopo-Molina and Arnold, 1991) carried out competition studies which showed that the heterodimeric p55/p61 HIV-1 RTase had a single template binding site, despite the fact that both monomers contain the fingers, palm and thumb domains. This suggests that the two subunits of the heterodimer act together in order to bind the DNA:RNA duplex.

This concept was further elucidated by structural studies of both MMLV RTase and HIV-1 RTase; Das and Georgiadis (2004) found a partial crystal structure of MMLV RTase and compared it to the crystal structure of HIV-1 RTase. It was found that the monomeric MMLV RTase forms the binding cleft on its own, while the binding cleft in HIV-1 RTase is produced by the interaction between the two subunits. This variation was suggested to be the result of a 32 amino acid insertion at the end of connection domain in the MMLV RTase. This insertion changes the angle at which the amino acid chain enters the RNase H domain, meaning that while the monomeric RTase can form its own binding cleft, the dimeric RTase requires another subunit to form the binding cleft. (Das and Georgiadis, 2004)

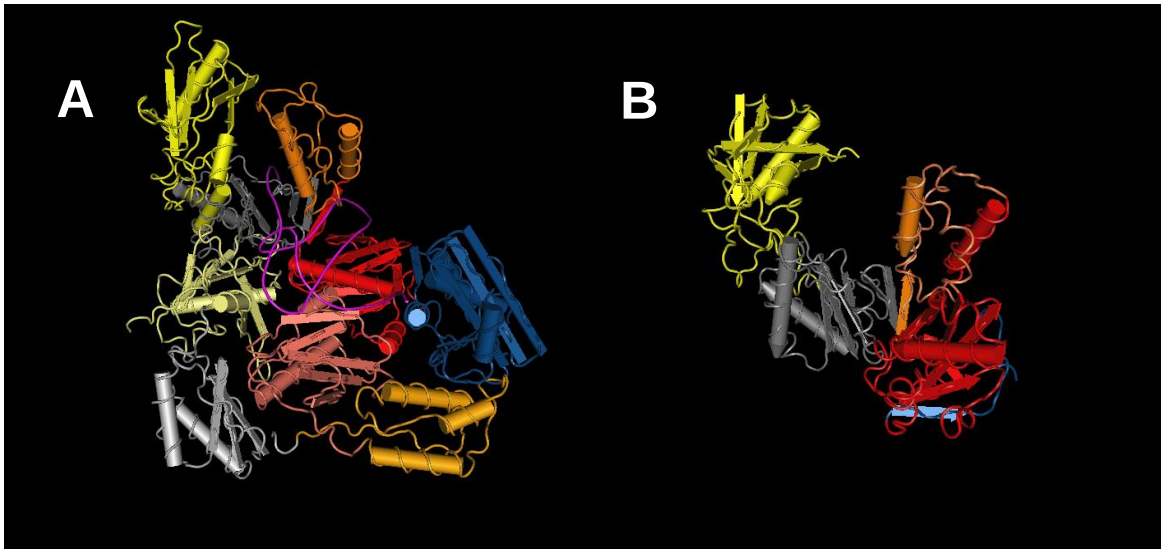


Figure 2: The crystal structure of reverse transcriptase from HIV-1 or MMLV. Fingers domain is shown in yellow; palm domain in grey; thumb domain orange; connection domain red; RNase H domain blue. (A) The crystal structure of HIV-1 reverse transcriptase reveals a hetero dimeric protein. The p66 domains are indicated with the colours listed above, while the p51 domains are indicated with lighter colours. (adapted from Jaeger et al., 1998) (B) The crystal structure of the N-terminal domains of MMLV reverse transcriptase. The full RNase H domain is not present in this crystal structure (adapted from Das and Georgiadis, 2004). Both images were obtained and adapted via the protein databank (Berman et al., 2000).

1.2.2 Functions of RTase

RTase must utilise multiple functions to efficiently reverse transcribe RNA to DNA. For example, the ssRNA viral genome of HIV-1 has been found be highly structured (Watts et al., 2009), meaning that RTase must be able to unwind RNA stem loops and other RNA secondary structures in order to efficiently carry out reverse transcription. Furthermore, one domain of RTase is a RNase H domain, indicating that the ability to cleave RNA is intrinsic to the activity of the RTase. This has been validated in studies wherein truncated RTase proteins lacking the RNase H domain were found to be less efficient at reverse transcription than full length RTases (Georgiadis et al., 1995).

In vivo, the enzyme relies both on the activity of the RTase, and on the structure and sequence of the viral genome. RTase utilises a tRNA primer, which binds to the primer binding site (PBS) near the 5' end of the RNA. DNA is elongated to the 5' end of the RNA, through the long terminal repeat (LTR) region of the viral genome. The length of DNA generated between the tRNA template to the 5' end of the RNA is known as the minus strand strong stop. The RTase then swaps template; the minus strand strong stop is also complementary to the 3' LTR of the RNA, allowing the DNA to be used as a primer for minus strand synthesis. The RTase then polymerises the minus strand of the DNA until it reaches the primer binding site, which is still bound to the tRNA primer. The RTase then dissociates from the minus strand and uses a conserved polypurine tract of RNA at the 5' side of the 3' LTR as a primer to make the plus strand strong stop. This plus strand strong stop is extended along the tRNA primer, until a methylated base in the tRNA prevents further elongation (Renda et al., 2001). The RNase H domain then cleaves the tRNA primer from the template, and a second template switch occurs. The plus strand of DNA is synthesised then using the negative strand of DNA as a template, via a circular DNA intermediate (Baltimore et al., 1979).

During *in vivo* reverse transcription, the enzyme must utilise all its different functions. For example, the cleavage of the tRNA primer from the minus strand DNA must use the RNase H activity of the RTase. Notably, the distance from the tRNA:DNA junction to the replication end point on the tRNA is 19 nucleotides (Renda et al., 2001), directly relating to the distance between the DNA polymerisation active site and the RNase H active site of 17-20 nucleotides (Schultz et al., 2009).

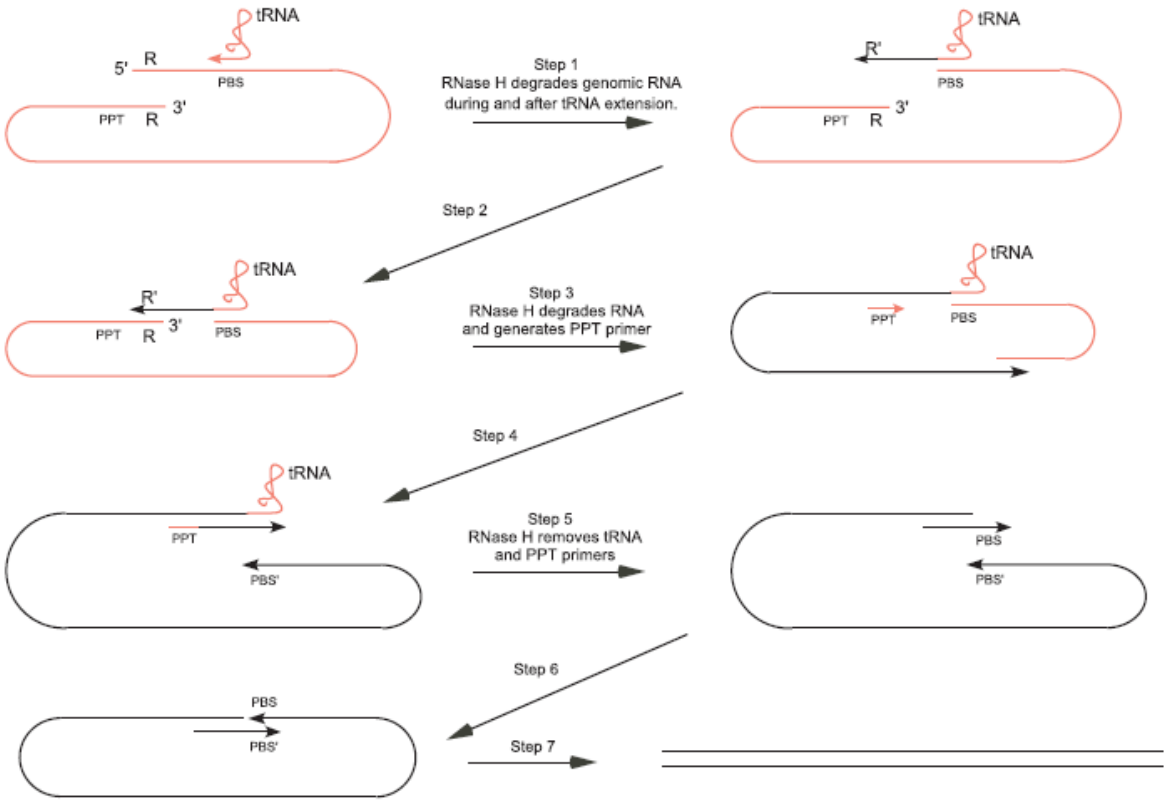


Figure 3: The mechanism of *in vivo* reverse transcription of a viral RNA genome (Champoux and Schultz, 2009) Red coloured strands indicate RNA, while black coloured strands indicate DNA. R and R' indicate a repeated region at each end of the viral genome while PPT stands for polypurine tract, a small oligo RNA sequence that is saved from degradation from the RNase H domain of the RTase. This PPT is then used as the primer in the second strand synthesis reaction. PBS indicates the primer binding site.

1.2.2.1 Substrate binding

The structure of the palm domain of RTase is similar to a ribonucleoprotein (RNP) motif. This structure is present in many RNA binding proteins and consists of 2 α helices and 4 β strands. These secondary structures fold to form a tertiary structure consisting of a antiparallel β sheet flanked at both sides by α helices. (Georgiadis et al., 1995)
Upon dNTP binding, the fingers domain of the RTase undergoes a conformational change, closing the cleft formed by the fingers and palm domains. In HIV-1 RTase, this conformational change acts to bring K65 and R72 into positions where they can coordinate the incoming dNTP, facilitating polymerisation using a 2 metal ion catalysis (Huang et al., 1998). Through mutational studies, it has been found that the K65R can increase the fidelity of the HIV-1 RTase, while reducing the rate of polymerisation catalysis. It was proposed that this decrease in catalysis is likely due to interactions between K65R and R72, wherein

stacking interaction are

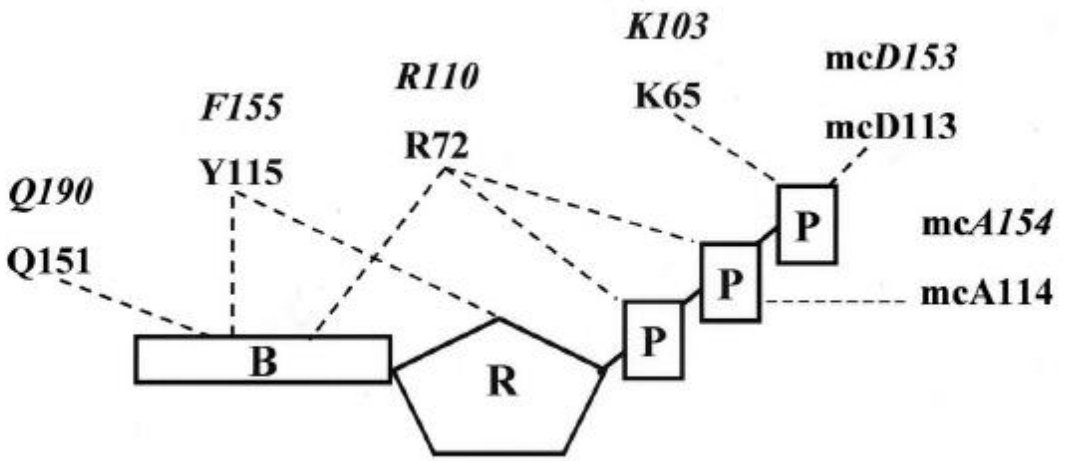


Figure 4: Schematic diagram illustrating the key amino acid side chains of MMLV and HIV RTase interaction with substrate dNTPs. The incoming dNTP is bound via electrostatic interactions with amino acids in the reverse transcriptase. Italicised amino acids are MMLV RT analogues while non-italicised amino acids are HIV-1 RT analogues. (Halvas et al., 2000)

likely to be formed, preventing the R72 from efficiently interacting with the β phosphate of the incoming dNTP. (Barrioluengo et al., 2011)

A single point mutation in MMLV RTase at F155V allows the RTase enzyme to incorporate rNTPs in the place of dNTPs. This mutation also confers a 2.8-fold decrease in fidelity (found using *lacZa* inactivation studies). Halvas et al., (2000) suggest that mutation prevents steric clashes between the aromatic group on phenylalanine and the OH group on the rNTP. This hypothesis was further validated by their own experiments, in which only the F155Y and F155W mutants of F155 resulted in a retrievable viral titre. The hypothesis is also backed by structural evidence, where the F155 analogue in HIV-1 RTase – Y115 - was shown to interact with the base of the incoming dNTP, and to be in close proximity to the ribose moiety of the dNTP. (Halvas et al., 2000)

1.2.2.2 DNA polymerisation

The mechanism by which RTase polymerises DNA from either an RNA template or a DNA template has been found to be similar to that of DdDps. Structurally, DdDps and RTases have a similar architecture; both enzymes have thumb, fingers and palm domains that

facilitate nucleic acid binding, dNTP binding and phosphatidyl transfer (Kohlstaedt L.A., et al., 1992).

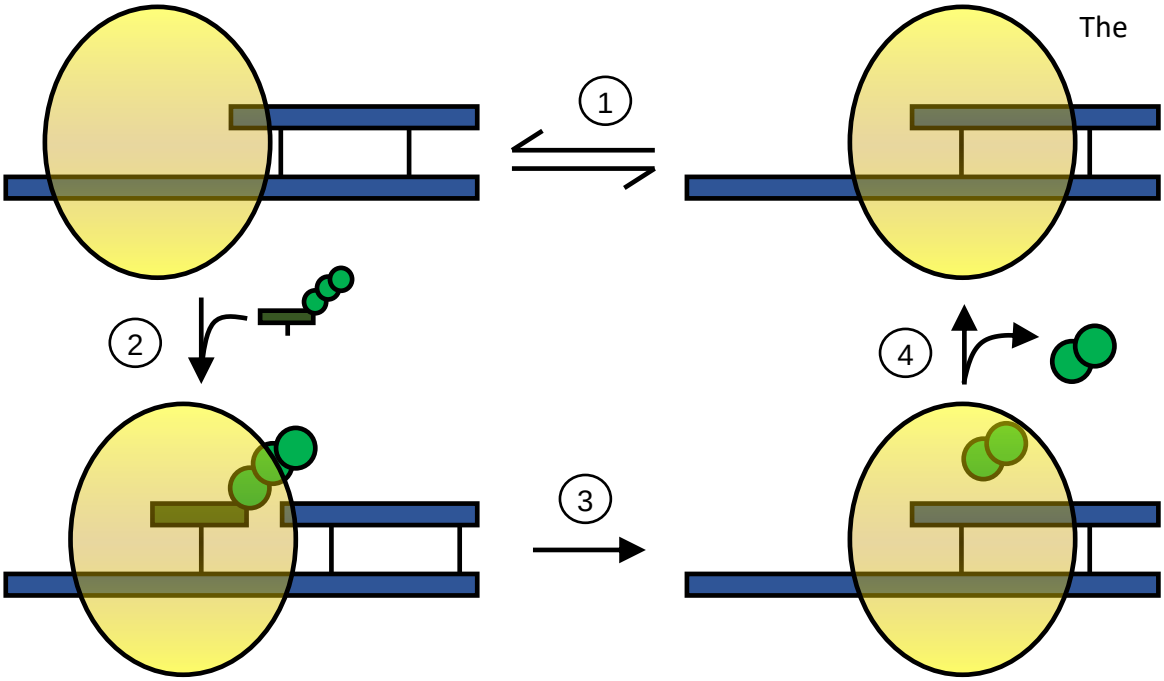


Figure 5: The processivity cycle of reverse transcriptase. (1) Reverse transcriptase diffuses along the nucleic acid. This can occur in either direction. (2) The binding of the correct dNTP causes a conformational change in the reverse transcriptase, trapping it in the post-translocation position. (3) The dNTP is incorporated into the growing DNA strand. This occurs via a two-metal ion mechanism. (4) Pyrophosphate is released from the reverse transcriptase, causing a conformational change back to the initial conformation.

polymerisation reaction occurs via a two-metal ion system, wherein two divalent metal ions coordinate and facilitate the polymerisation of DNA. The first metal ion (often denoted as metal A) acts to destabilise the 3'-OH of the DNA primer, facilitating the deprotonation of the hydroxyl group. This allows the 3'-O:$^{-}$ of the primer to carry out nucleophilic attack on the α phosphate on the incoming dNTP. The second metal ion (often denoted as metal B), stabilises the transition state by forming a pentacovalent structure with the α phosphate and the beta phosphate of the dNTP, which facilitates the removal of one oxygen group from the α phosphate, to the β phosphate. (Steitz, 1993)

1.2.2.3 Processivity

RTase moves along the RNA template in a processive manner. This movement occurs via a Brownian ratchet mechanism in which the binding of the dNTP complementary to the RNA is likely to be the rectifying reaction. Malik et al. (2017) looked at the movement of the RTase along a DNA template, as well as using hairpins to investigate strand displacement synthesis. Using a site-specific foot printing technique, they found that the pre- and post-translocation states of the RTase are at equilibrium and that the presence of a templated nucleotide can trap the RTase in the post-translocation state (Malik et al., 2017). This mechanism links back to the conformational changed in the fingers domain conferred by dNTP binding which is required for the efficient catalysis of polymerisation. (Huang et al., 1998; Steitz, 1993)

Crowther et al. (2004) found that the substitution of R116A in MMLV RTase resulted in a decrease in the processivity of the enzyme. After examining the crystal structure and binding kinetics of this mutant in comparison to the wild type RTase, they concluded that the R116A had a markedly decreased affinity for DNA duplexes, and that R116 is likely to be involved in hydrogen bonding interactions that are critical for DNA binding and the processive synthesis of DNA. (Crowther et al., 2004)

1.2.2.4 RNase H domain

RTase has a C-terminal RNase H domain. This domain acts to degrade RNA from DNA:RNA duplexes, and to produce primers such as the polypurine tract for second strand synthesis. The close proximal relationship between the RNase H and polymerisation domains may suggest that there are beneficial interactions between the two functional domains.

In a study by Zhan and Crouch (1997), it was found that the truncated RNase H domain of HIV-1 RTase and MMLV RTase retained activity, albeit at a lower level than the wild type, but lost specificity. This loss of specificity highlights that the linkage of the polymerisation domain and the RNase H domain is necessary for correct functionality.

One possible mechanism for this linkage is that pause sites in the reverse transcription of RNA are important to the specificity of the RNase H domain; RNA secondary structure induced pausing of the RTase could provide the time necessary for RNase H to cleave RNA. Zhan and Crouch (1997) found that the specificity of the RNase H for the tRNA:DNA junction

– a known pause point due to the requirement of template switching - was lost. This could provide some link between the secondary structure of the nucleic acid and the function of the protein (Zhan and Crouch, 1997).

1.2.2.5 Strand displacement synthesis

Unlike many other polymerases, RTase does not require a helicase to first unwind the nucleic acid in order for complementary strand synthesis to occur. Instead, RTase can use strand displacement synthesis to denature any double stranded nucleic acids while elongating the nascent 3' nucleic acid. This unwinding of the nucleic acid does not require any hydrolysis of ATP, suggesting RTase plays a less active role in strand separation than helicase (Whiting and Champoux, 1994). The exact mechanism for this strand displacement synthesis is still not fully elucidated, however, research suggests that it either occurs almost entirely by the thermal breathing of the nucleic acid (Malik et al., 2017), or by a combination of passive thermal breathing and interactions between the RTase and the nucleic acid duplex (Whiting and Champoux, 1998).

In a paper published in 1998, Whiting and Champoux measured the rates of strand displacement and non-displacement synthesis of RTase using quenching experiments. MMLV RTase was incubated with template DNA at 37°C such that viral derived DNA was extended either with or without a DNA duplex downstream of a radiolabelled primer. These reactions were quenched at various timepoints by the addition of formaldehyde and EDTA, and the lengths of the resultant DNA molecules were found by radio-imaging techniques. Whiting and Champoux found that strand displacement synthesis proceeded at a 3-4-fold slower rate than non-displacement synthesis. They additionally found that the effect of temperature on strand displacement synthesis was similar to that of temperature on non-displacement synthesis. This was posited as evidence that it is unlikely that strand displacement synthesis is entirely reliant on the thermal breathing of the nucleic acid. Kinetic analyses were then carried out which showed that the rates found in the experiments did not fit either a model where strand displacement was carried out entirely by thermal breathing, or entirely by interactions between the protein and the nucleic acid. Whiting and Champoux therefore suggested that strand displacement synthesis likely occurs by a mixture of thermal breathing and protein interactions.

The suggestion by Whiting and Champoux (1998) is contradicted by the work of Malik et al., (2017), wherein a molecule of DNA containing a hairpin-loop was tethered to two beads and measured the distance between the beads to find the rate of strand displacement synthesis. By varying the force applied to the beads, and the concentration of dNTPs added to the experiment, the dependency of strand displacement synthesis on these parameters could be elucidated. It was found that the RTase enzyme has a minimal effect on the stability of the replication fork, meaning that the thermal breathing of the nucleic acid is mostly responsible for the unwinding required by strand displacement synthesis. (Malik et al., 2017)

Whiting and Champoux (1998) investigated strand displacement synthesis by using both gapped and nicked templates, ultimately finding that a nicked template was not sufficient for efficient strand displacement polymerisation. This suggests that the endonuclease activity of the RNase H domain alone is not sufficient to facilitate generation of primers (such as the polypurine tract) in template switching, and that additional exonuclease activity must occur in order for second strand synthesis to occur in an efficient manner. This is because the RNase H domain of the RTase simply nicks the RNA template, whereas Whiting and Champoux showed that a nicked template is not sufficient to allow extension of the second strand. This means that there must be some helicase activity present in the system, or that a ribo-exonuclease acts to degrade the RNA from the RNA-DNA duplex after first strand synthesis is complete.

1.2.2.6 Terminal Transferase

In addition to the RdDp activity, RTase also has some level of terminal transferase activity; both avian myeloblastosis virus (AMV) RTase (Clark, 1988) and HIV-1 RTase have been shown to have terminal transferase activity. This ability to add untemplated nucleotides to the 3’ end of the DNA has been utilised in order to facilitate template switching *in vitro*; terminal transferase activity has been perturbed in order to introduce specific bases that can then be used as primers for second strand synthesis (Zajac et al., 2013; Zhu et al., 2001). These primers are essential in second strand synthesis, as in research labs, the RNA that is reverse transcribed is often not genomic viral RNA. This means it does not have the intrinsic features that facilitates template switching and second strand synthesis *in vivo*. Therefore,

in order to allow the synthesis of the second strand of DNA *in vitro*, DNA primers must be added. If the terminal transferase activity can be modulated in order to always introduce a certain sequence of nucleotides, a single primer can then be used, thereby increasing the proportion of full-length cDNA products that are ultimately amplified in comparison to the use of random primers.

The terminal transferase activity of RTases has been shown to be modulated by various nucleotide substrate analogues (Ohtsubo et al., 2017). Ohtsubo et al (2017) investigated the effect of seven compounds related to dNTPs to find if the terminal transferase "tailing" activity of MMLV RTase can be manipulated. These seven compounds included bases, ribonucleoside phosphates and deoxyribonucleoside phosphates. It was found that base analogues alone are not effective enhancers of the terminal transferase activity of MMLV RT; the tailing activity was more significantly enhanced by either ribonucleoside phosphates, or deoxyribonucleoside phosphates (enhancement was most prevalent upon addition of dNMPs). This suggests that either the phosphate moiety or the ribose moiety of the enhancer molecule is required for the enhancement of tailing activity. The same study also looked at the effect of different bases on the composition of the tail added to the 3' DNA, and it was found that in general, the addition of any base resulted in an enhancement of the complementary partner of that base being incorporated in the 3' tail. For example, addition of dGMP enhanced the tailing incorporation of cytosine. Moreover, the addition of both GMP and GDP as an enhancer led to a suppression of 3' guanine tailing.

The results of the Ohtsubo et al paper (2017) indicate a potential mechanism for the terminal transferase activity of MMLV RTase. The fact that base analogues alone are not as efficient at enhancing the tailing activity as ribonucleoside phosphates suggests that interactions between the RTase, and the template DNA ribose and phosphate moieties are required in addition to base pair interactions. This further suggests that the enhancer molecule binds to the MMLV RTase as if it were an additional nucleotide on the 3' end of the RNA, possibly utilising stacking interactions with the RNA template. This would allow the MMLV RTase to incorporate a dNTP opposite the transiently bound enhancer molecule as a 3' DNA tail.

1.2.3 Mutations of RTase

Since the discovery and realisation of the importance of RTase, there have been many attempts to mutate or evolve it. Some of these mutation experiments have been carried out in order to better understand the molecular biochemistry of the RTase, while others have been focused on improving a practical facet of the protein.

Site directed mutagenesis was used in 1987 by Larder et al. to identify functional regions of interest in order to better elucidate potential clinical pathways for the treatment of HIV-1. They identified six regions in HIV-1 RTase that tend to be conserved across RTases, as well as in other RNA polymerases. Amino acids in these regions were mutated using SDM, and each of the resultant mutants were expressed in *E. coli.* The mutants were assayed for RTase activity, and the concentration of two RTase inhibitors (phosphonoformic acid (PFA) and azidothymidine-triphosphate (AZT-TP)) at which 50% of the mutant RTase activity was abolished was found (ID_{50}). Using these observed changes in the ID_{50} value, as well as known modes of action of the inhibitors, Larder et al (1987) theorised that three residues might play a role in triphosphate binding, and that two out of these three might have an additional role in pyrophosphate exchange.

In 2009, Arezi and Hogrefe used a mutagenesis strategy combining random mutagenesis, and site directed mutagenesis to isolate a series of mutations in MMLV RTase that increased thermostability of the protein. Random mutagenesis was carried out using the low fidelity DNA polymerase, Mutazyme, from Agilent, which is reported to introduce 1-6 random mutations per kilobase of template after 20 cycles of EP-PCR based on the sequencing of 20 sample variants. After mutations were identified that positively impacted the thermostability of RTase at 55℃, site directed mutagenesis was used to carry out saturation mutagenesis at each of these positions, and at some surrounding positions. Ultimately, it was found that 5 mutations in concert (E69K/E302R/W313F/L435G/N454K) increased the activity of RTase at 55℃ by over 85%. This shows that there is a significant opportunity for making functional improvements in the RTase; while the wildtype MMLV RTase (RNase H-) has an optimal temperature of 45˚C, a targeted 5 mutations can increase this optimal temperature to 50˚C, with an increase in activity at higher temperatures.

1.3 Polymerase Chain Reaction

The polymerase chain reaction (PCR) is a universal feature of modern molecular biology research and diagnostic laboratories. Originally developed in 1985 by Kary Mullis (Saiki et al., 1985), PCR involves the use of a DNA polymerase (originally the Klenow fragment of bacterial DNA polymerase I) in order to amplify specific DNA sequences, as determined by a set of primers and template DNA. The discovery of a thermostable polymerase from *Thermus aquaticus* (Saiki et al., 1988) and subsequent incorporation into the PCR helped to increase the rate and specificity of PCR, allowing it to become a ubiquitous technique in molecular biology.

1.3.1 Quantitative Polymerase Chain Reaction

Quantitative PCR (qPCR) is the real time detection of DNA produced in a PCR. Originally developed using ethidium bromide – an intercalating agent that binds specifically to double stranded DNA – qPCR utilises a fluorophore and spectrometer to detect the amount of DNA at any stage of the reaction (Higuchi et al., 1992). This allowed the progress of a PCR to be tracked in real time for the first time. The process was later fully formalised by Heid et al. in 1996. Further developments came to the qPCR technique in the form of hydrolysis probes – short oligonucleotides that are cleaved by the exonuclease activity of Taq polymerase, thereby releasing a detectable fluorophore (Holland et al., 1991). These probes – also known "Taqman" probes, improved the level of confidence that an increase in fluorescence detected in a qPCR was due to the amplification of the specific gene or sequence of interest. RT-qPCR is an expansion of qPCR, utilising an RTase to create cDNA prior to a PCR, allowing RNA to be used as a template in the reaction. Originally, this technique simply used end-point PCR (Noonan and Roninson, 1988), and allowed relative quantitation of expression levels (Murphy et al., 1990), but was later adapted to incorporate qPCR (Gibson et al., 1996). Various mathematical models for predicting the progress of a qPCR have been developed in order to provide some level of absolute quantification of template levels. Early in literature, discrepancies arose between the comparative "unit" of qPCR, with threshold cycle (C_t), quantification cycle (C_q), Take Off Point (TOP), and crossing point (C_p), all in scientific parlance with similar or identical meanings. This time-point was often defined as the point at which the fluorescence exceeds the value equal to 10 times the baseline fluorescence

variation**.** This value was used in concert with other parameters, such as the amplification efficiency – estimated by an additional reaction – and the fluorescence at any cycle to estimate the starting fluorescence, which acts as a proxy for the starting template number. Improved mathematical modelling allowed estimation of amplification efficiency based on the experimental sample (Liu and Saint, 2002), and fitting of fluorescence to a curve thus allowing mathematical determination of a C_q parameter rather than a threshold value (Spiess et al., 2008)**.**

1.4 Random mutagenesis

Life requires a delicate balance of error and fidelity during the replication of genomic DNA. Mutations can lead to improved function of certain genes, allowing such genes in Nature to outcompete others and thrive in new niches, but other mutations can result in disease, loss of function, or even death of the host organism. As such, mutation rates *in vivo* are closely regulated, with replicative polymerases introducing mutations, and various cellular systems correcting or excising mutations. This rate of mutation introduction is also balanced alongside the size of the genome; species with larger genomes tend to have higher rates of mutation than species with smaller genomes resulting in fairly analogous rates of mutations per genome replication (Drake et al., 1998). Additionally, it has been theorised that the environmental niche in which a species lives and concurrent genetics of an organism will have an impact on the mutation rate of the organism. Gillespie (1981) posited that in a system with multiple alleles of a gene, selective pressure will act to either maximise the mutation rate, minimise the mutation rate, or to maintain the mutation rate at an intermediate value. Gillespie theorised that if one allele is vastly advantageous to the other, mutation will most often create a deleterious allele, and thus would be minimised. Equally, if all alleles are equally fit, the mutation rate would be maximised. Furthermore, selection would act to maintain an intermediate mutation rate under certain other conditions.

1.4.1 Error catastrophe

*In vivo, m*any viruses have high rates of mutation in comparison to those generally found in more complex organisms. For example, poliovirus is estimated to have a mutation rate of

0.03 mutations per synonymous site per year (Savolainen-Kopra and Blomqvist, 2010). This allows the virus to evolve at a very rapid rate (Jarvis and Kirkegaard, 1992), but brings them dangerously close to what is known as an error catastrophe. As a result, an effective drug against polioviruses – ribavirin – acts to increase the mutation rate of the virus, resulting in the death of the species (Crotty et al., 2001).

This error catastrophe was first theorised by Eigen and Schuster in 1977, wherein they stated that as a mutation rate increases from 0, a quasi-species is formed. This quasi-species acts as a species, but it composed of many different variants of the same master copy, with one or several variants dominating the quasi-species. At a certain mutational threshold, this quasi-species will be dispersed, as the "information" maintained by cohesive inheritance is "…dispersed over subsequent reproductions." This phenomenon was later modelled, showing that above the error threshold, the entire sequence space is covered. (Tarazona, 1992).

Error catastrophes do not dictate an upper limit on how much a gene can be mutated, but rather dictate how much a gene can be mutated in a single lifecycle while maintaining informational coherence. Additionally, experimental proof of this principle is limited to systems biology examples, wherein the mutation rate of an organism increased to the point that cohesion is lost. This is due to the fact that self-replication is a necessity of the error catastrophe system, however, some principles can be taken and applied to individual proteins.

1.5 Modelling PCR

There is a considerable body of literature published on the kinetic relationships between the concentrations of substrate, template and products during PCR reactions (see for example Boggy and Woolf, 2010). These studies have a direct bearing on the design and execution of qPCR experiments, in particular the requirement to determine the quantity of nucleic acid template present in the original sample being analysed, which is essential for quantitative transcriptomics (Chelly et al., 1988) and diagnostics (Bustin, 2000).

The modelling of EP-PCR has been explored previously: in 1995, Sun developed a model to explain the products derived in PCR, utilising the theory of branching processes (Watson and Galton, 1875). Sun introduced the concept of the "Generation" of a strand of DNA in a PCR.

This is a concept recognises that the amplified and mutated PCR products generated in EP-PCR provide alternative templates for subsequent template-mediated amplification reactions. Therefore, any mutations that occur are later amplified when a mutated product is used as the template. The "Generation" of any DNA strand is equivalent to the number of extension steps it has been through – i.e. the DNA strand generated from the original PCR template is of "Generation" one, as it has been through a single extension step (see figure 1.5). Importantly, the "Generation" of any strand is uncoupled from the cycle it was produced in; the "Generation" of a DNA strand is entirely dependent upon the "Generation" of the strand it used as a template. Thus, an equal number of first "Generation" strands of DNA will be produced in the first cycle as the last cycle of PCR. In the resultant model, Sun provided mathematical proofs that the probability that a randomly selected strand of DNA will be of any "Generation" will follow a binomial distribution (Sun, 1995).

In 2000 Moore and Maranas developed a model in order to predict the probability of obtaining a specific nucleotide sequence after a number of EP-PCR reactions (Moore and Maranas, 2000). In their model, they considered important points surrounding mutagenesis – namely that not all mutations have an equal probability of occurring, and that the sequence of a gene of interest would impact the probability of a mutation arising in said gene. Additionally, they recognised that this sequence-dependent mutation rate is not constant over the course of the PCR but will change based on the availability of individual free nucleotides. The Moore and Maranas (2000) model expands the work carried out by Sun, (1995), building on the concept of polymerase derived DNA strand "Generation" – here called the number of extension steps – which is better formalised in the context of EP-PCR. The "Generation" concept is then used in conjunction with the mutational matrix in order to provide a probability of a nucleotide *i* mutating to nucleotide *j* over the course of an EP-PCR. They additionally expanded this model to consider DNA shuffling and compared the resultant modelled data to previously reported experimental data.

Wang et al (2000) further developed the work of Sun (1995) to specifically consider the distribution of nucleotide sequences derived in EP-PCR. In order to estimate the mutation rate of a theoretical EP-PCR, Wang et al (2000) employed two different mathematical approaches in order to compare the number of base changes, and the number of pair-wise differences among a random sample of sequences. They also developed simulations in order to test their model against those of Weiss and von Haeseler (1997), and Sun (1995). These

simulations found that all models gave similar estimations of mutation rate when the mutation rate was low ($< 10^{-3}$ per base per PCR cycle), but that when the mutation rate was high, Weiss and von Haeseler's model and Sun's model underestimated the mutation rate, whereas the two methods carried out by Wang et al. (2000) approximated closer to the true mutation rate of the simulations (Wang et al., 2000; Weiss and Von Haeseler, 1997).

The model generated by Moore and Maranas (2000) was later adapted by Pritchard et al. (2005), to allow for non-optimal amplification efficiency, and variable amplification efficiency. Quoting data from real time PCR experiments, Pritchard et al. (2005) noted that the amplification efficiency of a PCR is almost never perfect, and that often results follow a sigmoidal or logistic curve, suggesting a systematic decline in amplification efficiency as the PCR proceeds. The reasons for this observed decrease in amplification efficiency were suggested as being a result of -

A: a decrease in the number of free nucleotides;

B: a proportional decrease in active enzyme compared to the number of DNA sequences;

C: a deterioration of the efficiency of the polymerase.

To this end, Pritchard et al. (2005) developed Python programs to model EP-PCR at: a constant, perfect amplification efficiency; a constant, imperfect amplification efficiency; and a variable, imperfect amplification efficiency. They then looked at the change in information - or Hamming distance - from the template sequence to sequences generated by their models, comparing the three models and finding the probabilities of obtaining a predetermined number of specified substitutions in a sequence using the models. They found that their model with variable amplification efficiency generated sequences that were closer to the template sequence than Moore and Maranas' original model (Moore and Maranas, 2000). This decrease in Hamming distance is likely due to the decreasing amplification efficiency in a PCR, resulting in "early-peaking" of the "Generation" of DNA strands in the EP-PCR, thereby resulting in a skewed population of amplicons in the final EP-PCR products. Pritchard et al., (2005) concluded by noting that changing the population limit (the threshold at which the amplification efficiency becomes zero), or altering the amplification efficiency could be a method for manipulating the number of mutations observed in an EP-PCR (Pritchard et al., 2005).

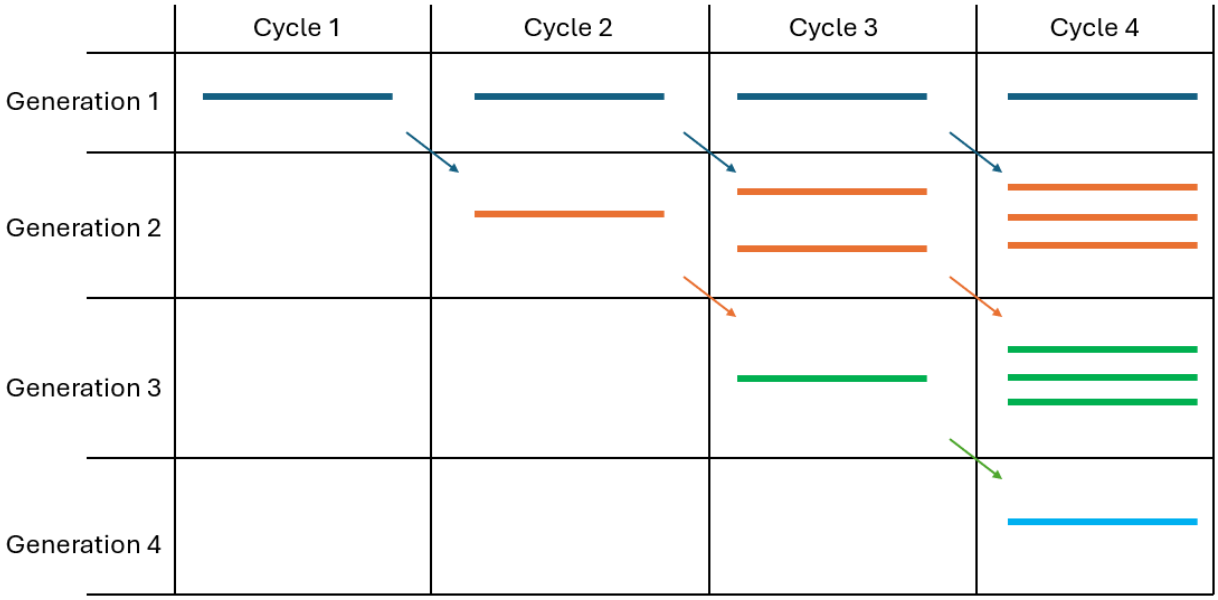


Figure 1.5: A diagrammatical demonstration of how the "Generation" number of a nucleic acid sequence is produced and varied in an EP-PCR. In this diagram, arrows represent replication of a nucleic acid sequence, creating a new sequence with a "Generation" score of one greater than the parental sequence. All sequences are retained in the reaction, and so can replicated again in future cycles of EP-PCR.

1.6 Aims of the study

In this study, we aim to elucidate a unified model for the introduction of mutations in an EP-PCR. Utilising a unique, error-prone polymerase (PhoEP), the model outlined in this thesis is compared to those of Moore and Maranas (2000), Wang et al. (2000), and Pritchard et al. (2005), in order to determine the optimal model to predict the number of mutations introduced over the course of an EP-PCR. These models will be used in concert with sequencing data from individual EP-PCR experiments in order to elucidate a likely per kilobase extended mutation rate for EP-PCR utilising Taq and manganese, or PhoEP. This work aims to build on existing literature surrounding modelling EP-PCR, and to provide some experimental data in order to support or reject the models and the associated theories.

The PhoEP protein will then be utilised in the production of RTase mutants, alongside other unrelated "control" templates to underpin the design of mutagenesis experiments, with the intention to improve thermostability of the RTase enzyme. The aim of this is to provide the part-funder of this thesis – Diagenode – a novel RTase to use in their diagnostic assays. These assays look specifically at low molecular weight, often non-coding, RNA molecules. As such, it is not possible to utilise poly(T) primers, and thus random primers must be used in the RTase stage of the RT-PCR. At low temperatures, these primers are particularly susceptible to primer dimer formation (Rychlik, 1995), and other features that might inhibit the RT-PCR. As such, it is theorised that a higher reverse transcription temperature might prevent inhibition of the reaction and might allow detection of a wider variety of small, non-coding RNAs.

2.0 Materials and Methods

2.1 Materials

2.1.1 Buffers, reagents and enzymes

All purchased buffers, reagents and enzymes were sourced from the suppliers in the following tables.

Equipment	**Supplier**
1.5ml Eppendorf tubes	StarLabs
0.2ml PCR tubes	StarLabs
15ml Culture tubes	Alpha Laboratories
20ml Universal tubes	Grenier BioOne
50ml Falcon tubes	Fisher Scientific
10µl Graduated tips	StarLabs
200µl Yellow tips	StarLabs
1000µl Blue graduated tips	StarLabs
P10 Pipetman® classic pipette	Gilson
P20 Pipetman® classic pipette	Gilson
P100 Pipetman® classic pipette	Gilson
P200 Pipetman® classic pipette	Gilson
P1000 Pipetman® classic pipette	Gilson
HiBind® DNA minicolumn	Omega
Arktik Thermal Cycler	Thermo Scientific
G:Box F3 Gel Imager	Syngene

1-14 Centrifuge	Sigma
3-16PK Centrifuge	Sigma
Powerpac HC	Bio-Rad
Wide Mini ReadySub-Cell GT Cell	Bio-Rad
WPA Lightwave S2000	BioChrom
NanoDrop 2000 UV visible spectrophotometer	Thermo Scientific
Heat Block	Geneflow
Scales	Geneflow
Open qPCR	Chai Bio
MinION	Oxford Nanopore
Flow cell (R9.4.1)	Oxford Nanopore
Mini-ProTEAN Tetra Cell	BioRad
Tetra Blotting module	BioRad
BioAnalyzer 2100	Agilent

Table 2.1.1.1: List of equipment and suppliers used in the course of this work

Reagent	**Supplier**
10x Standard Taq Reaction Buffer	New England Biolabs
10x Buffer for T4 DNA Ligase with 10mM ATP	New England Biolabs
10X Taq Mg-free standard buffer	New England Biolabs
dNTP solution	New England Biolabs
CutSmart® Buffer	New England Biolabs
3.1 NEBuffer	New England Biolabs
2.1 NEBuffer	New England Biolabs
Taq DNA Polymerase	New England Biolabs
Eco53kI	New England Biolabs
HindIII	New England Biolabs
NdeI	New England Biolabs
NotI	New England Biolabs
XbaI	New England Biolabs
BamHI	New England Biolabs
EcoRI-HF	New England Biolabs
T4 DNA Ligase	New England Biolabs
Quickload ® 100bp DNA ladder	New England Biolabs
Quickload ® 1kb DNA ladder	New England Biolabs
Hyperladder ® I	Bioline
Hyperladder ® IV	Bioline
pUC19 Vector	New England Biolabs

λ DNA	Fermentas
Nutrient Broth	Oxoid
Nutrient Agar	Oxoid
Agarose, Molecular Grade	Bioline
BugBuster® Protein Extraction Reagent	Novagen
$MgCl_2$	New England Biolabs
Monarch Plasmid Miniprep Kit	New England Biolabs
ISOLATE II PCR Purification/Gel Extraction Kit	Bioline
10000x SYBR Green	Sigma
Ligation sequencing kit (SQK-LSK109)	Oxford Nanopore
Flow Cell Wash Kit (EXP-WSH004)	Oxford Nanopore
Control expansion (EXP-CTL001	Oxford Nanopore
Flow Cell Priming kit (EXP-SLP002)	Oxford Nanopore
Thermolabile Exonuclease I	New England Biolabs
Quick Calf Intestinal Phosphatase	New England Biolabs
NEBNext Ultra II End repair/dA-tailing Module	New England Biolabs
NEBNext Quick Ligation Module	New England Biolabs
Agencourt AMPure XP beads	Beckman Coulter
SuperSignal™ West Pico PLUS Chemiluminescent Substrate	ThermoFisher
His-tag Antibody, pAb, Rabbit	GenScript
Anti-Rabbit IgG (H+L), HRP Conjugate	Promega
Agarose	Bioline

6X gel loading dye, purple	New England Biolabs
QuickChange SDM kit	Agilent
qPCRBio Probe 1-Step	PCRBio
OneTaq RT-PCR	New England Biolabs
SuperScript II	ThermoFisher
SuperScript III	ThermoFisher
Mini-PROTEAN Precast gels	BioRad

Table 2.1.1.2: List of reagents and suppliers used in the course of this work

2.1.2 Buffers

All buffers were made to the specifications outlined in the following table

Buffer	**Composition**
SDS Lysis Buffer	0.5% Sodium Dodecyl Sulphate; 50 mM Tris-HCl (pH = 8.0); 1 mM DTT
Taq Storage Buffer	50 mM KCl; 1 mM DTT; 0.1 mM EDTA; 50 mM Tris-HCl (pH = 7.6); 0.5 mM PMSF; 50% Glycerol
Stacking Polyacrylamide Gel	3.75% Acrylamide (37.5:1); 0.375 M Tris-HCl (pH = 6.8); 0.1% Sodium dodecyl sulphate; 0.1% Ammonium Persulfate; 0.1% TEMED
Resolving Polyacrylamide Gel	10% Acrylamide (37.5:1); 0.125 M Tris-HCl (pH = 8.8); 0.1% Sodium dodecyl sulphate; 0.1% Ammonium Persulfate; 0.1% TEMED
50X TAE	2 M Tris; 1M Acetic acid; 50 mM EDTA
Ampicillin	100 mg/ml in ddH_2O; working concentration 100 μg/ml
Kanamycin	50 mg/ml in ddH_2O
Chloramphenicol	25 mg/ml in EtOH
Ni-NTA Lysis/Binding buffer	50 mM NaH_2PO_4; 300 mM NaCl; 10 mM imidazole; pH 8.0
Ni-NTA Wash buffer	50 mM NaH_2PO_4; 300 mM NaCl; 20 mM Imidazole; pH 8.0
Ni-NTA Elution Buffer	50 mM NaH_2PO_4; 300 mM NaCl; 250 mM imidazole; pH 8.0
20X TBST	400 mM Tris-HCl; 1.5 M NaCl; 2% TWEEN-20 (v/v)

Western Blot Transfer Buffer	20 mM Tris; 150 mM Glycine; 20% Ethanol
2X SDS Loading dye	100 mM Tris-HCl (pH 6.8); 4% SDS; 20% Glycerol; 2% B-mercaptoethanol; 25 mM EDTA; 0.04% Bromophenol Blue
LB	1% Tryptone (w/v); 1% NaCl (w/v); 0.5% yeast extract (w/v)
LB Agar	1% Tryptone (w/v); 1% NaCl (w/v); 0.5% yeast extract (w/v); 1.5% agar (w/v)

Table 2.1.2: List and composition of buffers made in the laboratory

2.1.3 Bacterial Strains and plasmids

The name, genotype, and source of all of the major bacterial strains used in the course of this work can be found in the table below.

Strain	Genotype	Source
NEB5α	*fhuA2Δ(argF-lacZ)U169 phoA glnV44 Φ80Δ(lacZ)M15 gyrA96 recA1 relA1 endA1 thi-1 hsdR17*	New England Biolabs
BL21(DE3)	*fhuA2 [lon] ompT gal (λ DE3) [dcm] ΔhsdS* *λ DE3 = λ sBamHIo ΔEcoRI-B int::(lacI::PlacUV5::T7 gene1) i21 Δnin5*	New England Biolabs
Rosetta	*F-ompT hsdSB(rB- mB-) gal dcm (DE3)* pRARE *(CamR)*	Merck
GroESL	*F- lambda- ilvG- rfb-50 rph-1* pGro7 *(CamR)*	Professor SJ Foster

Table 2.1.3.1: List, genotype and source of all strains used in the course of this work

Plasmid	Description
pUC19	Cloning vector containing ampicillin resistance gene and pMB1 high-copy origin of replication. Also contains *lacZa* gene allowing for blue/white selection.
pET28a	Expression vector containing kanamycin resistance gene and pMB1 high-copy origin of replication. Also contains *lacI* gene, T7 promoter and terminator and f1 origin to facilitate expression via the T7 lysogen system
pJF010	Expression vector containing ampicillin resistance gene and pMB1 high-copy origin of replication. Also contains RTase antisense from a lac promoter.
pJF012	Expression vector based derived from pET28a. Contains RTase insert in multiple cloning site.
pQIS207	Expression vector containing kanamycin resistance gene and pBR322 origin of replication. Also contains NiFe gene under control of a T7 promoter.
pQIS257	Expression vector containing amplicillin resistance gene and a pMB1 high-copy origin of replication. Contains the *bom* (basis of motility) from pBR322, and expresses Rop protein to maintain low copy number. Contains the gene for PhoEP polymerase under control of a T7 promoter.
pPETase	Expression vector based derived from pET28a. Contains NiFe-PETase insert in multiple cloning site.

Table 2.1.3.2: List and description of all the major plasmids used in the course of this work

2.2 Methods

2.2.1 Purification of plasmids from bacterial stocks

Plasmids were purified from bacterial stocks using the Monarch Miniprep kit from New England Biolabs according to the manufacturer's instructions. A single bacterial colony was used to inoculate 3 ml LB containing the appropriate antibiotic and was grown overnight at 37^{o}C in an orbital shaker. The culture was retrieved and centrifuged at 17000 x g for 3 minutes. The supernatant was discarded, and the resultant pellet was resuspended in 200 µl Resuspension Buffer (NEB). 200 µl Lysis Buffer (NEB) was then added to the resuspended pellet, which was then gently inverted numerous times until the suspension was clear. 400 µl Neutralization Buffer (NEB) was then added to the suspension, which was inverted several times, and incubated at room temperature for 2 minutes, or until a white precipitate formed. The sample was then centrifuged at 17000 x g for 10 minutes, and the resultant supernatant was moved to a spin column (NEB). The spin column was centrifuged at 17000 x g for 1 minute, and the flowthrough was discarded. 200 µl Wash buffer 1 (NEB) was applied to the spin column, which was then centrifuged at 17000 x g for 1 minute. 400 µl Wash buffer 2 (NEB) was applied to the column, which was once again centrifuged for 1 minute at 17000 x g. The flowthrough was discarded, and the spin column was centrifuged dry for 1 minute at 17000 x g. 20-80 µl ddH_2O was applied to the column. After 2 minutes incubation at room temperature, the spin column was centrifuged at 17000 x g for 1 minute, and the flowthrough, containing the plasmid, was stored at -20 ℃.

2.2.2 Production of competent *E. coli* cells

E. coli of various strains -including BL21(DE3), GroESL, and NEB5α- were made competent to allow for transformation with a plasmid vector. Individual colonies of the desired strain were selected from a streak plate and inoculated into 3 ml LB. This primary bacterial culture was grown overnight at 37 ℃ with shaking at 200 rpm. The following morning, 250 µl primary culture was added to 25 ml LB in an Erlenmeyer flask, which was incubated at 37 ℃ with shaking at 200 rpm, until the OD_{600} exceeded 0.5. The culture was then transferred to a falcon tube and was incubated on ice for 30 minutes. The culture was then centrifuged at 5445 x g at 4 ℃ for 10 minutes, and the supernatant was discarded. The resultant pellet was then resuspended in 10 ml 0.1 M $CaCl_2$ and was incubated on ice for 30 minutes. The cells

were then centrifuged at 5445 x g at 4 ℃ for 10 minutes, and the supernatant was discarded. The pellet was resuspended in 0.1 M $CaCl_2$/15% glycerol, and 100 µl aliquots were stored at -80 ℃ until required.

2.2.3 Transformation of competent cells

Plasmid vectors were used to transform competent cells. Competent cells were retrieved from -80 ℃ and thawed on ice. 1-10 µl plasmid vector at a concentration ranging from 100 pmol/µl to 100 ng/µl was added to the cells, which were then gently mixed. The cells were incubated on ice for 30 minutes and were then heat shocked at 42 ℃ for 30 seconds. 400 µl LB was added to the competent cells, and the cells were allowed to recover at 37 ℃ with rotation for 1 hour. Transformants were then plated onto LB agar plates containing the relevant antibiotic.

2.2.4 Expression of recombinant protein

Expression strain *E. coli* - namely BL21(DE3), GroESL, and Rosetta - were used to overexpress various recombinant proteins. Following transformation with a specific plasmid containing the gene of interest, an expression strain *E. coli* was inoculated from a streak plate into 3 ml LB containing the relevant antibiotic and was grown at 37 ℃ with shaking at 200 rpm overnight. The next day, 250 µl primary bacterial culture was inoculated into a 250 ml Erlenmeyer flask containing 25 ml LB with the appropriate antibiotic. This secondary bacterial culture was grown at 37 ℃ until the OD_{600} was approximately 0.5, at which time the expression of the gene was induced using 1 mM IPTG. The induced culture was incubated at either 25 ℃, 30 ℃ or 37 ℃ for 1-4 hours; the temperature and length of induction depended on the optimised expression of the protein. Following induction, the secondary culture was centrifuged at 5445 x g for 10 minutes at 4 ℃. The supernatant was discarded, and the resultant pellet was either stored at -20 ℃ or was immediately lysed to extract the protein of interest.

2.2.5 Protein extraction

Various protein extraction protocols were followed over the course of this research; both sonication and chemical lysis were utilised in order to extract proteins from the cells. In most cases, the aim of the protein extract was to prevent denaturation and degradation of protein in the extract.

2.2.5.1 Sonication

Cell pellets were resuspended in 5 ml per gram (wet weight) of pellet in an appropriate buffer (often potassium phosphate buffer (see table 2.1.2)). The resuspended cells were incubated on ice for 30 minutes and were then sonicated at 40% amplitude for 6 bursts of 10 seconds, with 10 seconds break between each burst. The lysed cells were centrifuged at 17000 x g for 30 minutes, and the supernatant was separated from the pellet. All samples were stored on ice or at 4˚C.

2.2.5.2 Chemical lysis

BugBuster Protein Extraction Reagent (see Table 2.1.1.2) was used to lyse cells. The cell pellet was weighed and resuspended in 5 ml BugBuster reagent per gram of cell pellet. The resuspended cells were then incubated at room temperature for 1 hour with rotation. The cell lysate was then centrifuged at 17000 x g for 10 minutes. The resultant supernatant was separated from the pellet, and both fractions were stored at 4 ℃ until required.

2.2.6 *in vitro* transcription and translation

In cases where overexpression of a protein in a bacterial strain was inefficient, *in vitro* transcription and translation (IVTT) was used to express the protein of interest. The NEBExpress kit (NEB) for expressing genes utilising the T7 expression system was used. In this system, S30 extract is used to transcribe and translate a gene under the control of a T7 promoter. Reactions were set up according to table 2.2.6, and were incubated at 37 ℃ for 2-4 hours, with shaking at 200 rpm. IVTT reactions were then retrieved and stored at 4 ℃ until required.

Reagent	**Volume**
S30 extract	12.5 µl
Buffer	25 µl
RNase Inhibitor	1 µl
T7 polymerase	1 µl
Plasmid/amplicon	125 ng
ddH_2O	Up to 50 µl

Table 2.2.6: Volumes of reagents used in IVTT reactions.

2.2.7 SDS-PAGE

Sodium dodecyl sulphate polyacrylamide gel electrophoresis (SDS-PAGE) was used to visualise the proteins from cell lysates or other sources. Polyacrylamide gels were either bought (see table 2.1.1.2) or were prepared in the laboratory (see table 2.1.2). Samples were added to 2X SDS protein loading dye (see table 2.1.2) and were heated at 100˚C for 5 minutes. The samples were then centrifuged at 17000 x g for 5 minutes, and the supernatant was loaded onto the polyacrylamide gel. The SDS-PAGE was run at different constant voltages and lengths of time based on the desired resolution but was generally run at 180 V for 45 minutes.

2.2.8 Western blotting

Western blots are often used to validate the identity of a protein band obtained via SDS-PAGE. Following electrophoresis, the gel was submerged in western blot transfer buffer and incubated at room temperature for 15 minutes with shaking at 60 rpm. Filter paper, nitrocellulose membrane, and sponges were submerged in transfer buffer prior to the assembly of the western blot. The blot was assembled in the equipment by layering a 3 x 1 mm filter papers on a sponge and layering the polyacrylamide gel on top of that. The nitrocellulose paper was then layered on top, followed by 3 further 1 mm filter papers and the top sponge. This was then inserted into the equipment, and the proteins were transferred at 120V for 60 minutes. During this time the gel and filter were kept cool using ice. The membrane was then retrieved and was blocked by submerging it in TBST/2% (w/v) milk powder for 45 minutes at room temperature with shaking at 60 rpm. The membrane was then retrieved and submerged in TBST for 5 minutes with shaking at 60 rpm. The membrane was then submerged in TBST/2% (w/v) milk containing the primary antibody and was incubated overnight at 4 ℃ with shaking at 60rpm. The following day, the membrane was retrieved and was washed thoroughly in 1X TBST at room temperature for 10 minutes with shaking at 60rpm. This wash step was repeated 3 more times. After the wash, the membrane was submerged in 1X TBST/2% milk containing the secondary antibody and was incubated at room temperature for 60 minutes with shaking at 60rpm. The membrane was then once again washed 3 times in 1X TBST, for 10 minutes with shaking for each wash. The membrane was then retrieved, and the antibodies were visualised using SuperSignal West Pico PLUS in a chemiluminescent visualiser.

2.2.9 Ni-NTA column chromatography

Cell lysates and IVTT results were run on an Ni-NTA column in order to purify His-tagged proteins of interest. 50% Ni-NTA resin slurry (Qiagen) was added to an empty column and allowed to settle. The column was unstoppered, and the storage solution was allowed to flow out the column. The column was then equilibrated using 4 resin volumes of potassium phosphate buffer (see table 2.1.2), which was allowed to flow through the column. One resin volume sample was loaded onto the column, was allowed to flow through the column and captured in the “sample load” fraction. Eight resin volumes of Ni-NTA wash buffer (see

table 2.1.2) was loaded onto the column, and four fractions of two resin volume were captured in “wash” fractions. Six resin volumes Ni-NTA elute buffer (see table 2.1.2) were finally applied to the column, and six fractions of one resin volume were captured as “elute” fractions. The resin was then resuspended in Ni-NTA elution buffer, and stored alongside the other fractions at 4 ℃. All samples were then run on an SDS-PAGE gel in order to see which ones had the band relating to the protein of interest.

2.2.10 Reverse Transcription

Reverse transcription was used to generate cDNA that could be subsequently amplified by PCR. Reactions were generally set up according to the manufacturer’s instruction (see table 2.2.9), using a provided RTase and buffer. In some cases, an expressed and purified RTase was used in the reverse transcription reaction, in which case, the commercial RTase was substituted for 1-5 µl of the purified RTase sample. If random primers were used, prior to addition of the RTase, the RNA and primers were incubated at 65 ℃ for 10 minutes, after which time, the RTase was added, and the reverse transcription reactions were incubated at 25 ℃ for a further 10 minutes prior to the start of the reverse transcription. The reverse transcription reactions were incubated at a set temperature (designated by the manufacturer, see table 2.2.10) for 1 hour, and were subsequently heated at 80 ℃ for 5 minutes in order to inactivate the RTase. cDNA samples were then stored at -20 ℃ until required.

RTase product	SuperScript II		SuperScript III		OneTaq RT-PCR		qPCRBio Probe 1-Step	
Manufacturer	**ThermoFisher**		**ThermoFisher**		**NEB**		**PCRBio**	
Buffer	First Strand Synthesis Buffer (5X)	4 µl	First Strand Synthesis Buffer (5X)	4 µl	2X MMLV Reaction Buffer	10 µl	4X Mastermix	5 µl
RTase	SuperScript II	1 µl	SuperScript III	1 µl	10X MMLV Reaction Buffer	2 µl	20X UltraScript	1 µl
dNTPs	10 mM each	1 µl	10 mM each	1 µl	N/A	0 µl	N/A	0 µl
DTT	0.1 M	1 µl	0.1 M	1 µl	N/A	0 µl	N/A	0 µl
Primers	Random Hexameric Primers	2 µl	Random Hexameric Primers	2 µl	Random Hexameric Primers	2 µl	Gene specific primers (10uM)	1-2 µl each
	Gene Specific Primers (5uM)	2 µl each	Gene Specific Primers (5uM)	2 µl each	Gene Specific Primers (5uM)	1 µl each	Probe (10uM)	0.25-1 µl
ddH_2O	N/A	Up to 20 µl	N/A	Up to 20 µl	N/A	Up to 20 µl	N/A	Up to 20 µl
Reaction Temperature	42 ℃		55 ℃		42 ℃		55 ℃	

Table 2.2.10: The reagents and respective volumes used in various reverse transcriptase reactions.

2.2.11 Polymerase chain reaction

Polymerase chain reactions (PCRs) were carried out routinely for multiple different reasons: to amplify DNA for use in subcloning or IVTT; to randomly mutate genes of interest; to assay a polymerase for activity; or to confirm a specific nucleic acid sequence is present in a sample. Generally, the composition of the PCR stayed constant (see table 2.2.11), although the temperatures and lengths of time for each step of the amplification varied between samples, depending on the melting temperature of the primers and the length of the amplicon respectively. For a full list of the primers used, see appendix 1. The volume of polymerase and template added varied depending on the activity or concentration of specific preparations.

Reagent	**Concentration**	**Volume**
Primers	5 μM	2 μl each
Buffer	5X	5 μl
$MgCl_2$	25 mM	1 μl
Polymerase	N/A	1-5 μl
dNTPs	1.25 mM each	2 μl
Template	Variable	1-5 μl
ddH_2O	N/A	Up to 25 μl

Table 2.2.11: The reagents and respective volumes that were used in PCR experiments.

2.2.12 Reverse Transcription quantitative Polymerase Chain Reaction

Reverse transcription quantitative polymerase chain reaction (RTqPCR) was used in order to assess the catalytic capacity of RTase mutants. Two different RTqPCR methodologies were used: either dye-based or probe-based.

The dye-based RTqPCRs utilised a nucleic acid binding fluorophore in order to track the concentration of double stranded DNA in the reaction (see Chapter 1.3.1). Generally, Luna Universal One-Step RT-qPCR kits were for RTqPCRs. The reaction set up can be seen in Table 2.2.12. The RNA and primers used in dye-based RTqPCR experiments was total yeast RNA alongside Act1 primers (see appendix 1.1).

In probe-based RTqPCRs, fluorescence is generated by the hydrolysis of "Taqman" primers – generically known as hydrolysis probes (see Chapter 1.3.1 for details). The "qPCRBIO Probe 1-Step Virus Detect" kit from PCRBiosystems was primarily used over the course of this work. The reaction composition can be seen in Table 2.2.12.

In all RTqPCR experiments, unless otherwise stated, the reaction proceeded by incubating the mixture at 55 ℃ for 5 minutes to carry out the RT, followed by 3 minutes at 95 ℃ to inactivate the RTase. The reaction was then cycled between 95 ℃ for 15 seconds followed by 60 ℃ for 30 seconds for 50 cycles. The fluorescence was measured at the end of the 60 ℃ step each cycle.

	Dye-based RTqPCR		**Probe-based RTqPCR**	
Manufacturer	**NEB**		**PCRBiosystems**	
Buffer	Luna Universal One-Step Reaction Mix (2X)	10 µl	PCRBIO Probe 1-Step Virus Detect mix (4X)	5 µl
Enzyme	Luna WarmStart® RT Enzyme Mix (20X)	1 µl	UltraScript RTase (20X)	1 µl
Primers	Gene Specific Primers (10 µM)	1 µl each	Gene Specific Primers (10 µM)	1 µl each
Template RNA	Yeast Total RNA	1 µl	Synthetic COVID E gene (8000 copies/µl)	1 µl
Probe	N/A	0 µl	COVID E gene probe (10 µM)	1 µl
ddH_2O	N/A	Up to 20 µl	N/A	Up to 20 µl

Table 2.2.12: The reagents and respective volumes that were used in RTqPCR experiments

2.2.13 Site directed mutagenesis

Site directed mutagenesis (SDM) was used to change individual nucleotides in DNA sequences. QuikChange Site-directed Mutagenesis kits **(**Agilent**)** were used to make these mutations. A reaction was set up with 5 µl reaction buffer, 2 µl template plasmid, 1.25 each primer (100 ng/µl), 1 µl dNTP mix, 1 µl PfuTurbo DNA polymerase, and was made up to 50 µl with ddH_2O. The following thermal cycling reaction was then carried out in a thermocycler.

Stage	**Cycles**	**Temperature**	**Time**
1	1	95 ℃	30 s
2	18	95 ℃	30 s
3	18	55 ℃	60 s
4	18	68 ℃	120 s

Table 2.2.13: Thermal cycler program utilised for SDM reactions.
After thermal cycling was completed, 1 µl DpnI enzyme (Agilent; 10 units/µl) was added to degrade the plasmid template, and the reaction was incubated at 37℃ for 1 hour.

2.2.14 Agarose gel electrophoresis

Agarose gel electrophoreses were carried out in order to visualise the size and quantity of any DNA species present in a sample. Generally, 1% (w/v) agarose gels were made by melting 1g Agarose (Bioline) in 100ml 1X TAE. After it had cooled to approximately 50˚C, ethidium bromide was added to the molten gel to a final concentration of 1 µg/ml which was mixed thoroughly. The gel was then poured into a prepared gel plate, and a comb was inserted, depending on the quantity and capacity of samples required. The gel was then allowed to set. 1-25µl samples were mixed with 6X Gel loading dye, which were then loaded into the comb wells. Generally, the agarose gels were run at 100V for 45 minutes.

2.2.15 Agilent Bioanalyzer 2100

In cases where precise quantification and identification of DNA species was required, an Agilent Bioanalyzer 2100 was used along with the DNA 7500 kit and chip. In these cases, a gel-dye mixture was filtered and prepared following the manufacturer's instructions. This gel-dye mixture was then loaded into the Agilent DNA 7500 chip following the

manufacturer's instructions. 5 µl marker (supplied by Agilent) was loaded into all sample wells of the chip. 1 µl DNA sample was then loaded into each sample well, and 1 µl DNA ladder (Agilent) was loaded into the ladder well. The fully loaded chip was then vortexed for 60 seconds at 2400 rpm and was then loaded into the Agilent Bioanalyzer 2100. The DNA 7500 chip was run using the Agilent “2100 expert” software (Agilent, 2022).

3. Development of a technique for the controlled introduction of random mutations into recombinant nucleic acid sequences

Abstract

Random mutagenesis is a powerful experimental technique for investigating the relationship between gene sequence and function. The use of intrinsically error-prone DNA polymerases, or the addition of alternative cofactors and nucleotide substrates, have facilitated the development of high throughput screening techniques, enabling researchers to explore the "sequence space" associated with specific gene function. However, as random mutagenesis is inherently stochastic, it can be hard to predict the number of mutations that will be introduced by any of the current, random mutagenesis experimental strategies. It would therefore be advantageous to have a method of predicting the probability of obtaining a given number of mutations in a random mutagenesis experiment.

In this Chapter, theoretical models are developed in order to validate the data obtained from the experimental mutagenesis of a recombinant plasmid encoding a modified thermophilic rubredoxin fused to a sequence encoding a nickel binding domain from the human transcription factor p54. Several models are presented for the prediction of the average number of mutations expected to be introduced via amplification of the coding sequence with an error-prone DNA polymerase (in this case a variant of the replicative polymerase from *P. horikoshii*). Following the generation of the random mutants via error-prone amplifications, individual clones were sequenced, and the number of mutations analysed alongside the predictions made using the different theoretical models. Simulated data are also developed in order to provide more robust statistical information. The overall aim of this Chapter is to select the model providing the closest fit to the simulated and experimentally observed mutation frequencies, in order to inform the design of random mutagenesis experiments using this experimental approach.

3.1 Introduction

Since the advent of qPCR, several authors have attempted to rationalise the frequency of random mutations arising using error-prone amplification strategies (Bakhtina et al., 2007; Boggy and Woolf, 2010; Pritchard et al., 2005; Spiess et al., 2008). These studies have been significantly influenced by the simultaneous advances in quantitative PCR (qPCR) and the

analogous RT-qPCR). These technological advances simplify the quantification of the rate of nucleic acid amplified by inspection of the PCR reaction curves, generated due to the change in fluorescent intensity in the reaction. Fluorescence change and detection may be achieved by the inclusion of a nucleic acid intercalating agent that, when bound to dsDNA shows concomitant increase in fluorescence (Higuchi et al., 1992). Alternatively, a fluorophore may be incorporated into an oligonucleotide sequence which is cleaved and released upon extension of the DNA (Holland et al., 1991), thereby providing a differential fluorescence signal.

The measurement of the fluorescent reporter – be it a hydrolysis probe, or intercalating agent – acts as a proxy for the determination of the relative yield of the DNA amplicon in the qPCR. The point at which a meaningful determination of amplicon yield can be made is influenced by the sensitivity of the instrument used. Historically, this point is referred to as the C_t (threshold point), C_p (crossing point), or TOP (take off point), however since the publication of a series of recommendations regarding the publication of qPCR experiments (Bustin et al., 2009), it is commonly known as the C_q, or quantification cycle. C_q was originally suggested to be the value that is 10 times greater than the standard deviation of the baseline fluorescence (Heid et al., 1996).
Using such data, by plotting the observed C_q against template DNA concentration in control samples, it is possible to fit the qPCR data to a standard curve, and thus estimate the starting concentration of DNA in an unknown sample. To do so, there are many different fitting models (Liu and Saint, 2002; Pfaffl, 2001; Spiess et al., 2008) that estimate the starting concentration based on different parameters and assumptions. Early fitting models were broadly split into 2 categories: absolute and relative quantification. Relative quantification does not provide information regarding the number of original template sequences, but rather allows comments to be made about the increase or decrease of template sequences in comparison to one another, or itself under different conditions**.** Absolute quantification relies on the mathematical modelling of a qPCR reaction and fitting various parameters to the experimental qPCR curve. Early absolute quantification of qPCR required a calibration curve to determine the amplification efficiency of the qPCR reaction with different known concentrations of initial template and assumed a constant amplification efficiency**.** More recent absolute quantification equations are more complex,

and can predict a changing amplification efficiency based on the data available above the C_q value (Liu and Saint, 2002). These modern fitting equations assume a sigmoidal curve, with either four (Alexander et al., 2004), or five parameters (Spiess et al., 2008) that can be changed to fit the curve to the data.

The modelling of EP-PCR adds an additional challenge for the modelling of qPCR, in that another parameter must be considered; the mutation rate of the EP-PCR system is a vital parameter in the modelling of EP-PCR. Furthermore, while modelling of qPCR does not consider the generation of each resultant amplicon, it is clear that this is a key factor in modelling EP-PCR (Sun, 1995).

Previous models of EP-PCR have identified the concept of generation – or the number of extension steps that any PCR product is the result from. This was first identified in 1995 (Sun, 1995), and was later formalised in 2000 (Moore and Maranas, 2000). Later work into EP-PCR models identified the imperfect nature of the amplification efficiency as an as of yet unmodelled phenomenon, which would have a distinct effect on the distribution of generation in the final EP-PCR products (Pritchard et al., 2005). Much of the work carried out focusses either on the probability that PCR products will be unmutated after an experiment (Wang et al., 2000), or the probability that a specific mutation will be obtained after a reaction (Moore and Maranas, 2000).

The model outlined by Moore and Maranas (2000) (known as the Moore model going forward) identified that the probability of any sampled sequence being the result of any number of extension steps (or being of a generation) is given by a normal distribution $Z_{N,n} = 2\binom{N}{n}$ wherein $Z_{N,n}$ is the number of sequences of generation *n* after *N* cycles of PCR. This equation was further adapted by a model outlined in Pritchard et al., (2005) (known as the Pritchard model going forward), to include a metric for imperfect amplification efficiency: $Z_{N,n} = S_0\lambda^n\binom{N}{n}$, where S_0 is the starting number of strands of DNA in a PCR, and λ is the amplification efficiency of the PCR. They also then expanded this concept to include a variable amplification efficiency that decreases linearly to 0 as the number of sequences in the PCR increases towards an arbitrary maximum number of sequences (Pritchard et al., 2005).

The Moore model also makes note that the mutation rate of the EP-PCR system might change as the reaction proceeds. They identify that different mutations have different

probabilities of occurring, and quote a per-cycle mutation rate matrix containing each possible mutation. Additionally, they identify that for example, the probability of an A being mutated to a T is dependent on whether that particular A has been mutated to any other nucleotide in the EP-PCR so far. As such, the probability that a specific mutation will occur follows a recursive relationship which is dependent on the probabilities of all previous generations (Moore and Maranas, 2000).

Pyrococcus horikoshii is an archaea originally found at hydrothermal vents on the ocean floor by González et al. (1998). In its natural habitat, *P. horikoshii* is exposed to temperatures exceeding 90˚C and high pressures**.** The cellular machinery of *P. horikoshii* must therefore survive and function at these high temperatures, making the polymerase ideal to use in PCR. A polymerase from a different *Pyrococcus* species - *P. furiosus* - has been used in PCR since the 1990s (Lundberg et al., 1991) as the 'Pfu' polymerase. In 2004, Biles and Connolly mutagenized the polymerase from *P. furiosus* in order to generate a low-fidelity mutant that could be used in EP-PCR. They found that two mutations -D215A and D473A- decreased the fidelity of Pfu polymerase 500-fold, increasing the rate of mutation from $1.4x10^{-6}$ to $7x10^{-4}$ per kilobase. It was posited that these mutations inactivated the exonuclease proofreading domain of Pfu polymerase, and interrupted the hydrogen bonding pattern of water molecules between two alpha helices, thereby conferring greater flexibility and allowing the polymerase to introduce more incorrect nucleotides (Biles and Connolly, 2004). These mutations were subsequently made in the polymerase from *P. horikoshii* by Professor David Hornby's lab (unpublished work), wherein the analogous mutations were found to have the same effect on fidelity. This error-prone polymerase from *P. horikoshii* (PhoEP) was used to introduce mutations in genes of interest.
While the approximate number of mutations that will arise following an EP-PCR using the PhoEP is known, the parameters of an EP-PCR that will give a predetermined desired number of mutations is not currently known. Multiple different simulations and models were produced in order to try and accurately predict how mutations will be introduced over the course of an EP-PCR, and how the parameters of an EP-PCR can be modulated in order to increase the probability of achieving a predetermined number of mutations.

3.2 Iterative computational modelling of EP-PCR

3.2.1 Binary rate-based simulation

An initial model was constructed using Python programming language. This model served as a starting point and allowed identification of the number of mutant strands of DNA compared to the number of wild-type or unmutated strands of DNA. This simple model acted in a loop process, with variables keeping track of the amount of total DNA at each cycle, the amount of wildtype and mutant DNA at the given cycle, and the amount of wildtype and mutant DNA in the previous cycle.

The program operates by doubling the number of the previous wildtype and mutant strands and subtracting the number of new mutants from the new wildtype number or adding the number of new mutants to the new mutant number. This number is determined by multiplying the number of new sequences in the wt column (equal to the prevwt value, as the program assumes perfect amplification efficiency) to a predetermined value of error rate. This error rate is defined as the proportion of sequences generated that would be mutated, and so represents the error rate per kb if the sequence is assumed to be 1kb in length. The program then "prints" the new values in a tabular format and proceeds to the next loop cycle, representing the next PCR cycle.

Once the predetermined number of cycles has been reached, the ratio of wildtype to mutant sequences is calculated and printed (see appendix 2.1). An example output of this program is displayed in table 3.2.1.

This program assumes many parameters, including that the amplification efficiency, remain constant and maximal, and that the error rate of the polymerase is itself constant. These assumptions are probably an oversimplification of the actual events taking place during amplification: for example, the rate of amplification is known to decrease as the PCR proceeds, and the error rate of the polymerase is known to vary based on base composition. However, the program defines a starting point for this work.

Cycle	Total DNA (strands)	wt	mut
0	2	2	0
1	4	3.8	0.2
2	8	7.225	0.775
3	16	13.74688	2.253125
4	32	26.17539	5.824609
5	64	49.87886	14.12114
6	128	95.12286	32.87714
7	256	181.5554	74.44464
8	512	346.8163	165.1837
9	1024	663.0806	360.9194
10	2048	1268.876	779.124
11	4096	2430.343	1665.657
12	8192	4659.292	3532.708
13	16384	8940.973	7443.027
14	32768	17173.92	15594.08
15	65536	33020.31	32515.69
16	131072	63551.48	67520.52
17	262144	122435.8	139708.2
18	524288	236120.8	288167.2
19	1048576	455833.6	592742.4
20	2097152	880902.5	1216250
21	4194304	1704121	2490183
22	8388608	3300084	5088524
23	16777216	6397373	10379843
24	33554432	12414505	21139927
25	67108864	24116059	42992805

Table 3.2.1: An example output of the "InitialBasicModel.py" found in appendix 2.1. The simplicity of the binary wildtype or mutant allowed the development of a satisfactory early model, generating a prediction of the proportion of wildtype (wt) to mutant (mut) amplicons that would be expected for a chosen number of cycles and polymerase fidelity. Additionally, the very low processing power required allowed the program to be developed and run on the most basic computers

3.2.2 Rate-based iterative computational model of EP-PCR (BasicStrandModel.py)

Building on the insights elucidated by the first model – "InitialBasicModel.py" – further computational models of EP-PCR were made in order to gain insight into how many mutations were made in any EP-PCR reaction. Once again, these computational models were made in Python programming language. In the initial, iterative rate-based model, a 2-dimensional matrix is made, wherein the *i* length is the number of cycles, and the *j* length is an arbitrary number of mutations. An arbitrary number is entered into the first cell of the matrix (0 cycles, 0 mutations) representing the initial number of strands of DNA in this

simulation. The model then proceeds by doubling all numbers in the first column into the second column.

For each cell in the second column, starting at the highest number of mutations (i.e. the bottom row), the number of sequences that would gain the number of mutations to reach the given resultant number of mutations is calculated using **equation 1.** This number of sequences is then subtracted from the active cell and added to the appropriate cell further down the column. This process is repeated for each possible number of mutations to be introduced up to the arbitrary mutation limit. Once each cell in the second column has been calculated, the program repeats the function by doubling each cell into the next column and calculating the number of mutations once again. This is repeated until the desired number of cycles of EP-PCR had been achieved (see appendix 2.2). A representation of this process is demonstrated in figure 3.2.2.

The results of this EP-PCR model appear to be inconclusive. The data predict a gaussian distribution of the number of mutations will be introduced over the course of the EP-PCR, however there are some issues with the program. These issues are namely that in a similar way to the model described in 3.2.1, this model allowed fractional numbers for quantities of different mutations. Furthermore, if a very large maximal number of mutations is chosen, the total number of sequences in the model does not increase as expected. This is because the higher numbers of mutations would pull a nominally small value from each preceding number of mutations. This nominally small value would be too small for Python to hold in its float-8 memory and so would be stored as 0. However, as many of these nominally small values would be subtracted from each number of mutations, the nominally small values would add up to a value that could be subtracted from a fewer number of mutations.

$$(1) \qquad probability\ of\ n\ mutations = x\mu^n$$

Where x = number of molecules of DNA

μ = probability of a single mutation occurring in replication of one molecule of DNA

n = number of mutations introduced per molecule of DNA

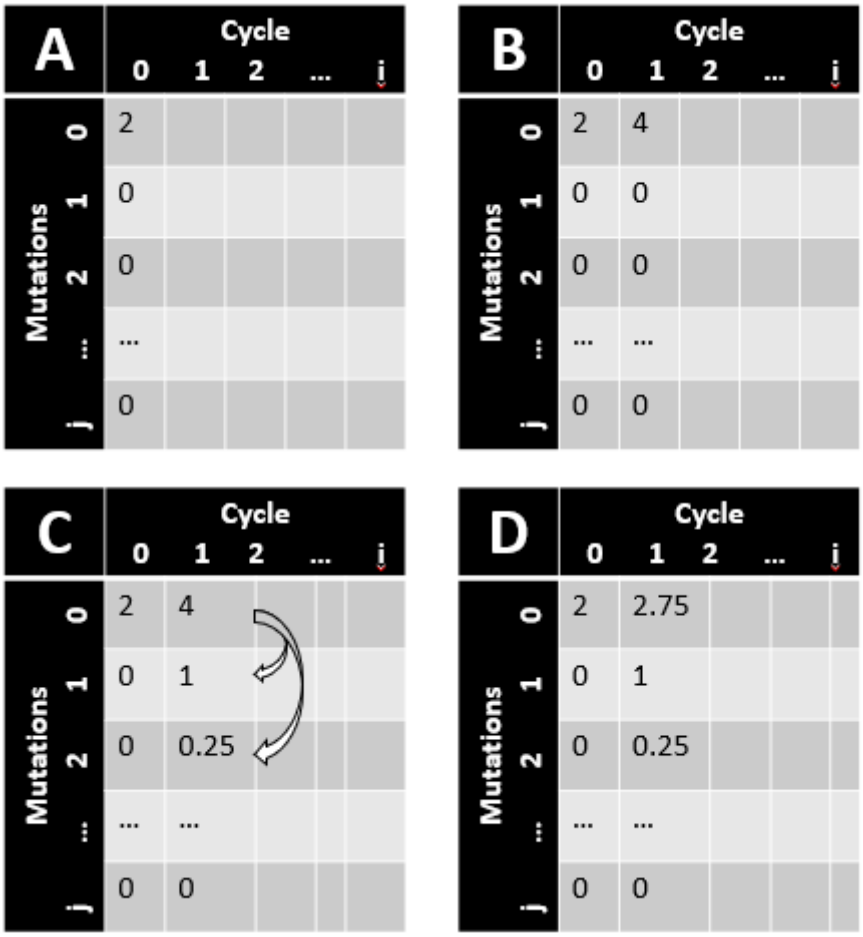


Figure 3.2.2: A representation of the rate-based iterative computational model of EP-PCR based on the Python program presented in appendix 2.2. In this representation, a mutation rate of 0.25/molecule DNA replicated is used. (A) An empty 2D matrix is created, and an arbitrary value is selected for the starting number of DNA strands. (B) The number of DNA strands of each number of mutations is doubled into the next cycle. (C) Starting with the highest number of mutations (j), the number of molecules of DNA that will be mutated is calculated and subtracted from each column cell. The number of molecules of DNA that will be mutated to result in each other number of mutations per molecule of DNA is then calculated and added to the appropriate cell.
In this representation, 1 molecule of DNA is added to the 1 mutation cell, as 1 = 4 * 0.25, and 0.25 is added to the 2 mutations cell from the 0 mutations box, as the probability that two mutations occur is 0.25^2 per molecule of DNA. (D) The sum of all new mutations is then subtracted from the working number of mutations – in this example, 1.25 is subtracted from the 1 mutation box, as the total number of mutations included in this example is low; in the program, mutations would occur successively up the number of mutations until the value of mutations was less than $2.25*10^{-308}$, as this is the smallest float possible in Python without any modules.

3.2.3 Iterative simulation of EP-PCR

In order to accurately predict the number of mutations introduced over the course of an EP-PCR, simulations of an EP-PCR were made in Python programming language. These simulations worked by generating a string of n length consisting entirely of 0s, representing a single molecule of DNA. This string was then replicated character by character, with a set probability that a "mutation" would occur, and the 0 would be converted to a 1. If a "mutation" occurred on a character that was already a 1, there was a 1 in 3 probability that the 1 would be converted back to a 0. The resultant string from this "replication" was then put into a list containing all other string sequences. All strings in this list of strings were then replicated again. This logic was cycled through until the desired number of EP-PCR cycles was achieved (see appendix 2.3.1).

The parameters of the EP-PCR simulation were varied; notably, the number of cycles and the probability that a mutation would occur were varied. Each set of parameters were repeated 100 times in order to ensure the data were accurate, and the resultant mean number of mutations for each repeat was determined. The mean and standard deviation of these means for each parameter pair was then calculated and plotted on a graph (see Figure 3.2.3). The Pearson's correlation coefficient for the mean and the standard deviation was calculated. A 2 tailed t-test was then carried out on the resultant t-ratios to find the P value for the correlation of the mean and the standard deviation to the number of cycles and mutation rate of the simulations.

It was found that as both the mutation rate and the number of cycles of the simulation increase, the mean number of mutations also increases ($1.54x10^{-13} < p < 3.78x10^{-10}$, and $1.11x10^{-14} < p < 1.26x10^{-9}$ respectively). Furthermore, as the mutation rate of the simulation increased, the standard deviation of the means of the reactions also increased ($p < 0.0001$). This significance was not constantly applicable to the increase in number of cycles, as the P value ranged from 0.003 to 0.25 with varying mutation rate.

These data suggest that a consistent predetermined single number of mutations cannot be achieved using EP-PCR as the resultant mean number of mutations introduced varied. As the introduction of mutations is ultimately stochastic, the number of mutations introduced in an EP-PCR will vary from reaction to reaction - even when keeping all parameters identical. It

does show that increasing the error rate of the PCR reaction will have a greater effect on the theoretical mean number of mutations introduced than number of cycles will.
It is therefore more appropriate to attempt to find a discrete probability mass function (PMF) in order to describe the probability that any given number of mutations will be introduced over the course of an EP-PCR. This PMF will then allow parameters to be optimised in order to maximise the probability of obtaining the desired number of mutations.

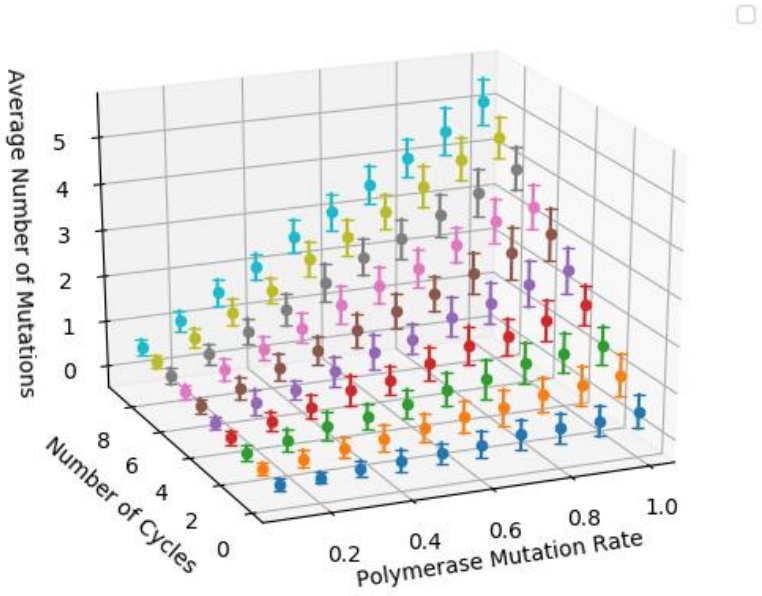


Figure 3.2.3: The results of the EP-PCR carried out by the program outlined in appendix 2.3.1. 100 replicates of each data point were used to find the mean and standard deviation of each cycle/mutation rate datapoint. These data were then plotted onto a 3D graph alongside the standard deviation. The data show that as the number of cycles or the polymerase mutation rate increases, so too does the average number of mutation in an EP-PCR.

3.2.4 Simulated EP-PCR

Following production of the iterative model of EP-PCR, a further iterative model was made in order to determine the impact of changing the mutation rate on the number of amino acid mutations that would be introduced over the course of an EP-PCR. This simulation program worked in a similar way to the binary simulation model, whereby for each nucleotide in each sequence, the Python module “random” would be used to create a

pseudo-random integer representing the chance of a mutational event occurring. Due to computers being incapable of producing truly random integers, pseudo-random integers must be used in all computation that relies on stochastic processes. These integers are produced using an algorithm from a seed value. This integer was used to determine whether a mutation event occurred at each of the positions in the molecule of DNA; if the integer was less than a predetermined threshold - corresponding to the fidelity of the simulated polymerase - a mutation event would occur. However, in this latter model, rather than simply increasing a 0 to a 1, the model would randomly mutate the nucleotide to any of the other nucleotides. This utilised a dictionary of dictionaries, wherein each of the 5 possible identities of nucleotide (A, C, T, G, and "-" representing a deletion) were the first level dictionary keys, and each had a value consisting of a second level dictionary containing all possible mutations with associated probability values. Upon a mutation event, a second pseudo-random integer between 0-100 would be generated to determine the identity of the mutation. The probabilities of possible mutations would be multiplied by 100 and added together until the sum of the probabilities exceeded the random integer. The dictionary key pertaining to the probability that was added to exceed the random integer was then determined as the mutation (see Figure 3.2.4 for an overview of the simulation process). The program carried out one cycle of an EP-PCR and saved the output to a fasta file. This allowed more cycles to be carried out before the RAM of the computer was saturated. The input file of sequences to be mutated was also a fasta file, inputted using the sys.argv command from the command line, allowing the program to be iterated for as many cycles as desired. In addition, the program saved the sequences with a fasta heading containing information such as the cycle in which the sequence was generated, and the generation number of the amplicon (see chapter 3.3.3). This allowed further analyses to be carried out on the sequences generated by this simulated EP-PCR.

The results of this secondary iterative simulation of EP-PCR showed that as the number of cycles increased, the average number of mutations increased (see figure 3.2.5b). The number of mutations introduced per cycle of PCR increased as the cycle number increased, however, when this value is normalised for the number of sequences - or length of information; related to "Generation" - replicated the number of mutations introduced per kb replicated stays fairly constant (data not shown). Furthermore, the proportion of amplicons with any number of mutations that is of any "Generation" follows a general

gaussian distribution, with the standard deviation increasing as the “Generation” number increases (see figure 3.2.5c).

The number of amplicons with any given “Generation” (see Chapter 3.3.3) was shown to follow a Gaussian distribution (see figure 3.2.5a), and the number of mutations introduced for any given generation was shown to follow a Gaussian distribution (see figure 3.2.5c). These data corroborate the statistical model of the introduction of mutations in EP-PCR as is explored in Chapter 3.3.

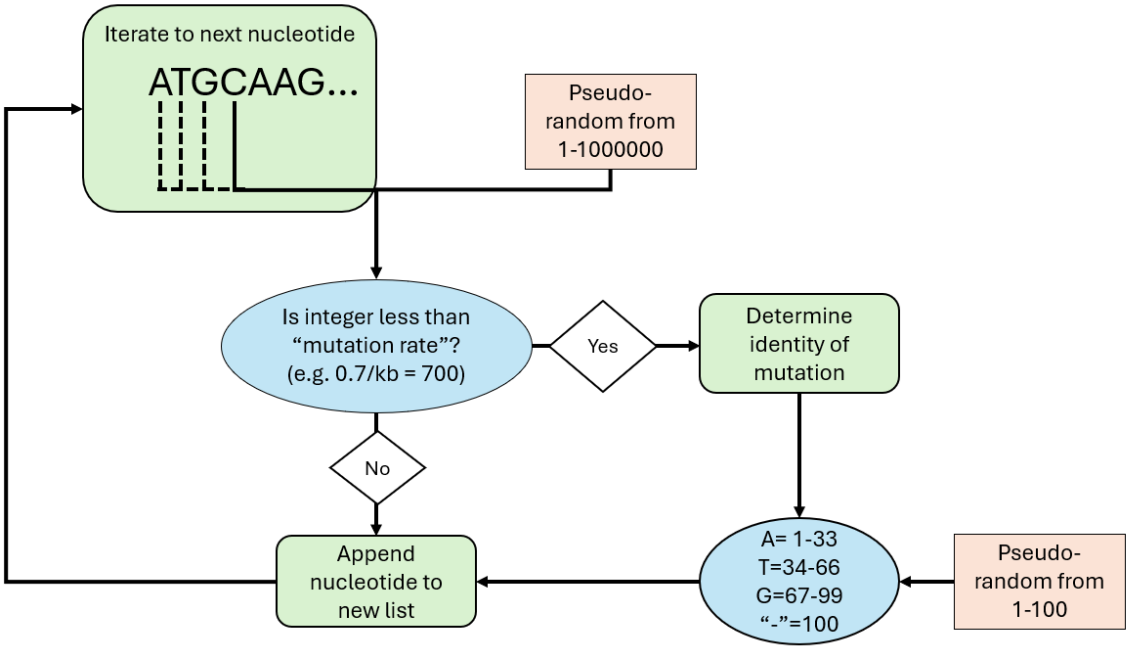


Figure 3.2.4: A diagrammatical representation of the process carried out by the simulated EP-PCR program. Note that pseudo-random integers were used in order to closely approximate the stochastic nature of EP-PCR.

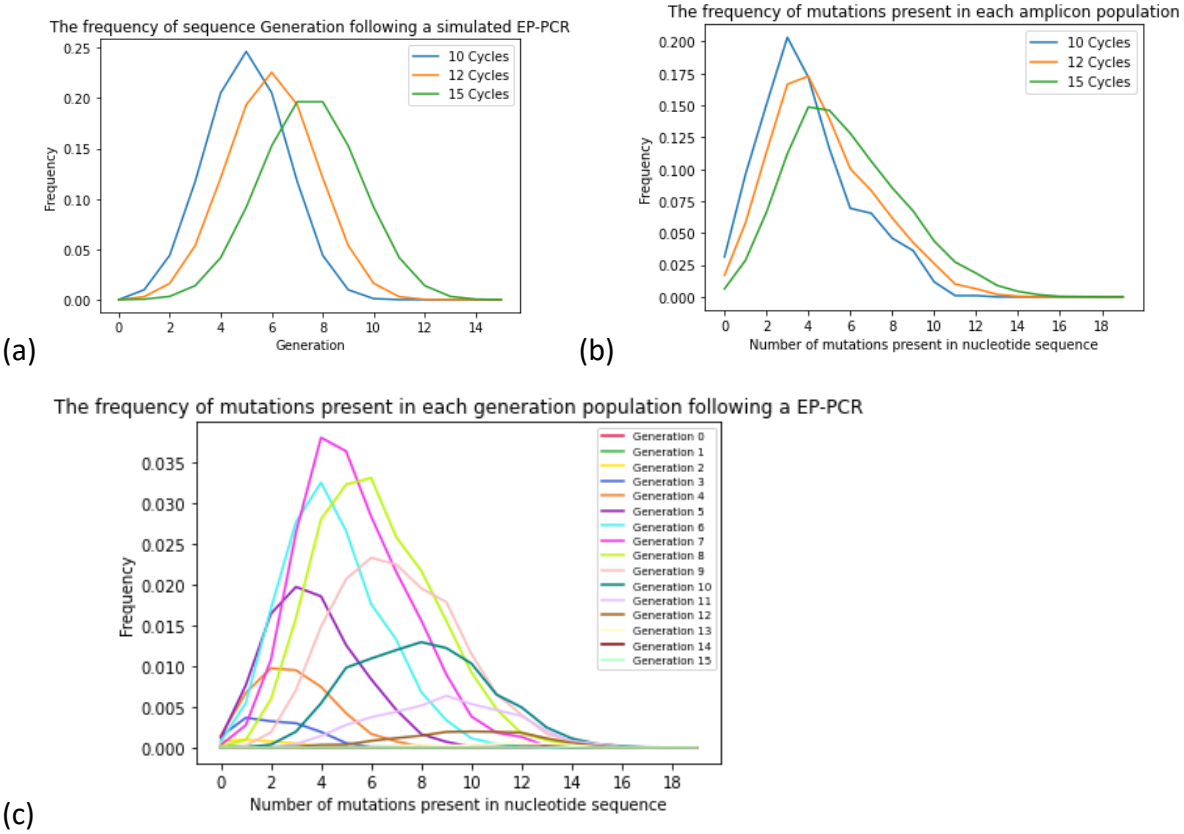


Figure 3.2.5: The frequency of generation number in a simulated EP-PCR, and the frequency of mutations and mutations per generation number following a simulated EP-PCR were calculated and plotted on a line graph. In all cases, the mutation rate of replication was set at 0.7 mutations per thousand simulated bases extended. The number of cycles simulated in (c) was 15.

3.3 Mathematical modelling of EP-PCR

As noted in the section 3.2, the introduction of mutations in an EP-PCR is a stochastic process, meaning that it is impossible to precisely predict. However, it is possible to calculate the PMF of the number of mutations that will be introduced, and therefore to vary parameters such that the desired number of mutations has the largest probability of occurring. As such, it is necessary to develop a PMF in order to calculate the probabilities that any given number of mutations will be introduced, for a predetermined number of cycles of EP-PCR at a pre-set rate of mutation.

3.3.1 Exponential distribution of introduction of mutations

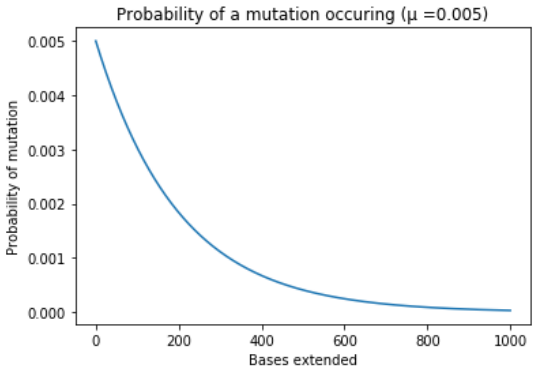
In Chapter 3.2 the rate of mutation has been considered to be similar to a rate constant - a set number that determines the proportion of molecule of DNA that will be mutated in any cycle of replication. As the modelling of EP-PCR has been recontextualised as a PMF, it stands to reason that the rate that mutations are introduced should be reconsidered too. In this probability-based context, the rate that mutations are introduced in an EP-PCR is represented by a PMF, wherein the number of bases polymerised since the previous mutation event, and the intrinsic fidelity of the polymerase are parameters that modulate the probability that any extended nucleotide will be mutated. **Equation 2** follows from this logic and is presented in figure 3.3.1.

EQUATION 2: Exponential distribution of frequency of introduction of mutation

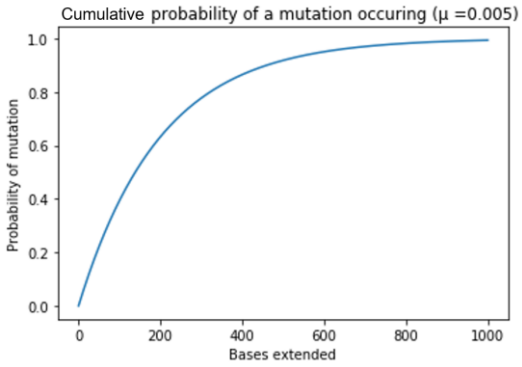
$$(2) \qquad \boldsymbol{Freq \sim Exp(\mu) = \mu e^{-\mu x}}$$

Where μ = mutation rate per kilobase extended

x = number bases extended (kb)



(a)



(b)

Figure 3.3.1: The probability densities of a mutation occurring in 1000 base pairs, for a given mutation rate of 0.005 (5 mutations per kb), using equation 2.

3.3.2 Poisson distribution of number of mutations introduced for a given length extended

Given that the probability of a mutation occurring is given by an exponential PMF in respect to the length of nucleic acid extended, it follows to reason that the probability that any given number of mutations that will be introduced for any set length of nucleic acid extended will follow a Poisson distribution. This means that it is possible to find the most probable number of mutations that will be introduced in a single cycle of PCR.

EQUATION 3: Equation of the Poisson distribution of number of mutations introduced

$$(3) \qquad mutations \sim Po(\mu l) = \frac{\mu l^n \,.\, e^{-\mu l}}{n!}$$

Where: n = number of mutations

μ= mutation rate (per kilobase extended)

l = length of DNA extended (kb)

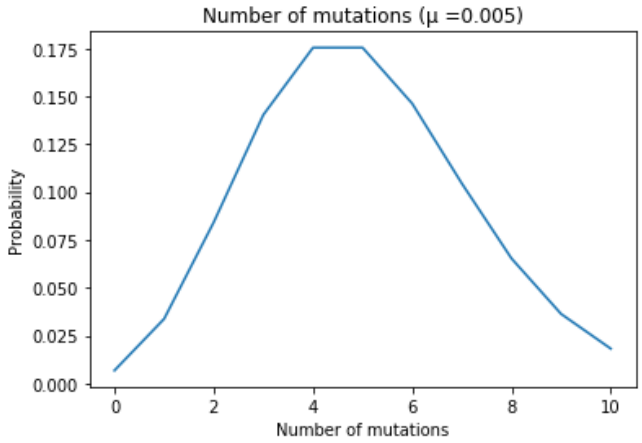


Figure 3.3.2: Probability mass function of number of mutations that will occur after 1000bases extended with a mutation rate of 0.005 (5 mutations per kb)

3.3.3 Binomial distribution of the length of generation of DNA strand

In an idealised PCR - wherein a single double strand of DNA is amplified exponentially by doubling each cycle - the first cycle of PCR only has a single possible template; that of the DNA inputted into the PCR. In the following cycles, the number of possible templates for the polymerase to replicate doubles each cycle; in the second cycle both the input DNA, and the DNA generated in the first cycle can be used as the template and replicated. This doubles again in the third cycle to include the DNA strands generated in the second cycle, and so on. Mutations accrued in earlier cycles of PCR will also be replicated - any mutations introduced in the first cycle of an idealised PCR will thus be present in 50% of the final DNA molecules. To represent this idea of "error carried forward", it is necessary to distinguish between the length of the DNA molecule from the length of template extended that results in each DNA molecule. For example, DNA that is generated from the original template inputted into the PCR is the result of a single extension step, whereas the DNA that is generated from the single extended DNA are the result of two extension steps. Crucially, this extension step count for each strand of DNA is unlinked to the cycle in which the strand of DNA was generated; the extension step count of each strand of DNA is entirely dependent on the extension step count of the strand of DNA that it used as a template. Therefore, there will be an equal number of one extension step DNA strands generated in the first cycle of PCR as there will be in the fiftieth cycle (due to there being the same number of zero extension step, or initial template strands of DNA).

The number of extension steps that a strand of DNA is the result of has a direct impact on the number of mutations that it is likely to carry. This is because while each strand of DNA is equal in length, each is the result of a different number of extension steps, meaning that the

information contained within the strand of DNA has been replicated an additional time. In a high-fidelity PCR, this would not matter, however in an EP-PCR, the method of replication introduces errors in a manner dependent on the length of DNA extended. To this end, while each strand of DNA is the same length, it can be said to be the result of a different length of information extended, with this length extended being directly correlated to the number of extension steps, and the length of the amplicon. For example, the strand of DNA generated from the original template DNA has an extension step count of one and is the result of the one amplicon length of DNA extended, while a DNA strand generated with an extension step count of two is the result of two amplicon lengths of DNA extended, despite only being a single amplicon length long.

In order to implement this concept in the mathematical modelling of EP-PCR, a new parameter must be introduced. This parameter represents the number of extension steps that any DNA strand is the result of, and is therefore directly correlated with, the length of information extended which results in any strand of DNA. This number of extension steps is also known as the generation of a DNA strand.

The probability that any DNA strand in the final library will be of any given generation will be dependent upon the amplification efficiency of the PCR system; in a perfect system, the amplification efficiency will be 0.5, and therefore the proportion of DNA strands generated in the preceding cycle will be 0.5. In more realistic PCRs, the amplification efficiency will drop below 0.5, as a PCR reaction is not truly exponential. Using this amplification frequency, the probability that any amplicon will be of any given iteration can be found using the PMF **equation 5**

Equation 4: The amplification efficiency of a PCR reaction assuming a constant amplification efficiency

$$(4) \qquad f = 1 - \left(\frac{startDNA}{totalDNA}\right)^{\frac{1}{c}}$$

Where c = the number of cycles

EQUATION 5: Binomial PMF of amplicon generation number

$$(5) \qquad generation \sim Bin(g,c) = \frac{c!}{g!(c-g)!} \times f^{g} \times (1-f)^{c-g}$$

Where: i = given iteration of amplicon

c = number of cycles in EP-PCR

f = amplification efficiency

g= generation number of amplicon

Equation 6: The length of genetic information extended as relates to generation and length of a particular amplicon

$$(6) \qquad l = g \times l_a$$

Where: l = length of genetic information extended

g = generation number of amplicon

l_a = length of amplicon (kb)

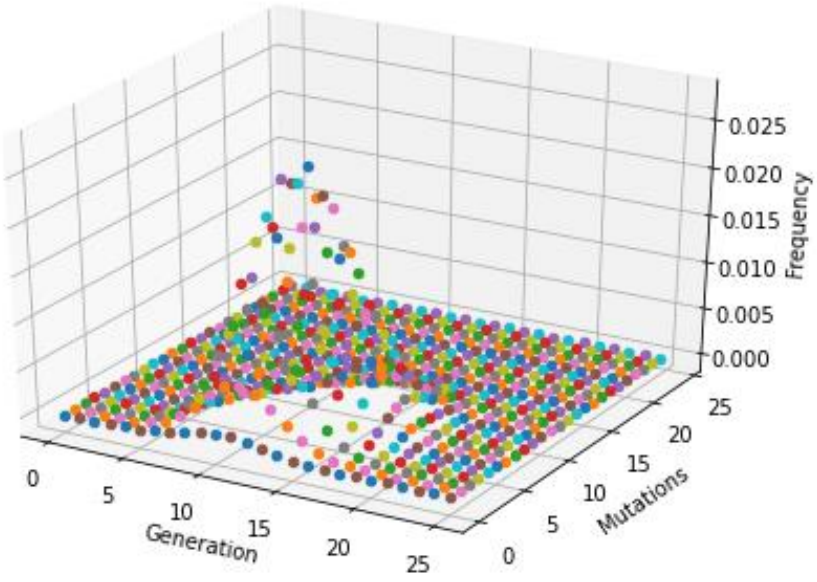
3.3.4 A probabilistic model of the introduction of errors in an EP-PCR

Using the previously defined equations and formulae, it is possible to find the overall probability that any number of mutations will be introduced in an EP-PCR. This model relies on the fidelity of the polymerase being known, and that the amplification efficiency of the PCR is known and does not change over the course of the reaction. The probability distribution for the generation of the amplicons can be found using the binomial distribution of generation for a given amplification efficiency (equation 4). This probability provides a predictor of the ratio of length of nucleic acid extended, which can be found by multiplying each generation by the length of the amplicon. Each length of nucleic acid extended is then used in a Poisson distribution (equation 3) to find the overall distribution of mutations for that generation. These values are then multiplied by the probability of the generation occurring to find the probability that any given number of mutations will be introduced for

any given generation of amplicon (see figure 3.3.4(A)). As the generation of the amplicon is irrelevant to the final library - and is indeed impossible to determine in a real library - the generation can then be eliminated, and the graph can be flattened to histogram (see figure 3.3.4(B)).

The data from this probabilistic model show a slightly skewed normal distribution, with more amplicons having a lower number of mutations than higher number of mutations - the mean number of mutations is generally slightly larger than the modal number of mutations. The data also further confirm the observation from the simulations - increasing either the number of cycles or decreasing the fidelity of the polymerase increases the mean number of mutations that will be introduced in an EP-PCR. Additionally, these data show that the standard deviation of the number of mutations introduced will also increase as both the number of cycles, and the error rate of the reaction increase.

(a)



(b)

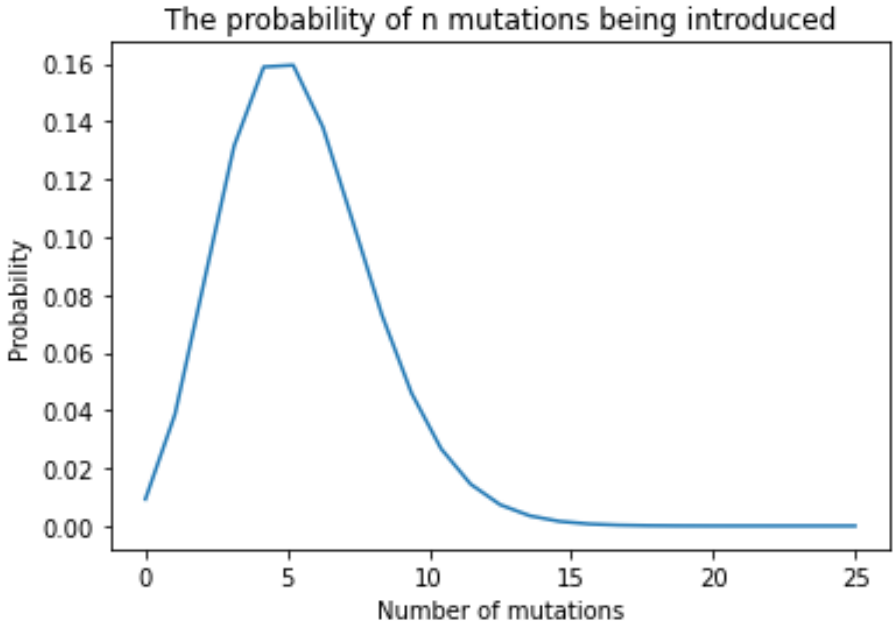


Figure 3.3.4: The probability of mutations being introduced over the course of an EP-PCR. (a) The probability distribution of the number of mutations that would be introduced over the course of a 25 cycle EP-PCR with intrinsic error rate of 0.4/kb DNA extended was found. First the distribution of generation was found using the binomial equation found in 3.3.3, which gave the proportion of each generation in the final theoretical product. Additionally, the distribution of number of mutations that would be introduced depending on the length of genetic information extended was calculated using the Poisson equation found in 3.3.2. These two distributions were multiplied together to find the probability of each generation/mutation pair occurring. **(b)** As the generation value of any amplicon is more theoretical and does not affect the amplicon beyond determining the probability of mutations being introduced, the generation can be ignored in many cases. As such, a graph depicting the number of mutations introduced regardless of the generation was produced.

3.3.5 Results

The probabilistic model provides an insight into how mutations are introduced over the course of an EP-PCR; the model allows a rapid determination of the expected distribution of mutations in a given EP-PCR for a given number of cycles and reaction fidelity.

There are clear limitations of the model that make it hard to prove in a laboratory setting. Primarily, the fact that the amplification frequency is fixed over the course of the EP-PCR is unrealistic. Studies have shown that as the number of cycles of PCR increases, the amplification efficiency progressively decreases (Peirson et al., 2003)**.** As such, to effectively use the model, it is necessary to either use an average amplification efficiency of the PCR - thereby underestimating the quantity of amplicons made in early cycles and overestimating the quantity of amplicons made in later cycles - or use the initial amplification efficiency, thereby greatly overestimating the quantity of amplicons generated in later cycles. This limitation could be minimised by experimental design; by running the PCR at a suboptimal extension and melting temperature, using all reagents in great excess, and by running the PCR for fewer cycles, it might be possible to minimise the deterioration of the amplification efficiency in a laboratory setting. By reducing the extension and melting temperature, the amount of polymerase that would be denatured over the course of the PCR would be minimised. Using all reagents in great excess would ensure that the kinetics of the PCR would stay fairly constant. Similarly, reducing the number of cycles of PCR would mean that fewer reagents are used and thus are kept in excess, thereby minimising the effect to the kinetics of the PCR.

The combined Binomial and Poisson model of the probability of n mutations being introduced is statistically identical to the results obtained from EP-PCR nucleotide simulation outlined in 3.2.4. This significance was shown by running a simulation of 10, 12 or 15 cycles of EP-PCR, with a mutation probability of 0.7 per kilobase extended, mimicking that observed by Biles and Connolly (2004). The number of mutations introduced in the simulation was then counted using the program “EPPCRAnalyse.py” (see appendix 2.6). This program iterated through the fasta file made by the simulation and counted the number of mutations in sequences of the same length as the original input sequence. If the sequences were of different length, the program first carried out a Needleman-Wunch algorithm to

pairwise align the sequence and the wildtype sequence, after which the number of mutations were counted. These data were then recorded in a 2D array, where the i component denoted the “generation” of the amplicon, and the j component denoted the number of mutations in the amplicon. A PMF of the binomial/poisson model of EP-PCR, utilising the same input parameters - rate of mutation, length of amplicon, and number of cycles - was then generated.

The probability of each number of mutations occurring was found for simulated and generated datasets and were plotted in graphs (see figure 3.3.5). These probability values were then multiplied by the total number of amplicons generated in each experiment. For 10 cycles, this was 1023, for 12 cycles 4095, and for 15 cycles 32767. These values were used in a paired t-test, to determine whether any differences between the data were statistically significant. In each case, it was determined that there was no significant difference between the data, with a t-value of 0.0581, 0.0019, and 0.0003 for 10, 12 and 15 cycles respectively.

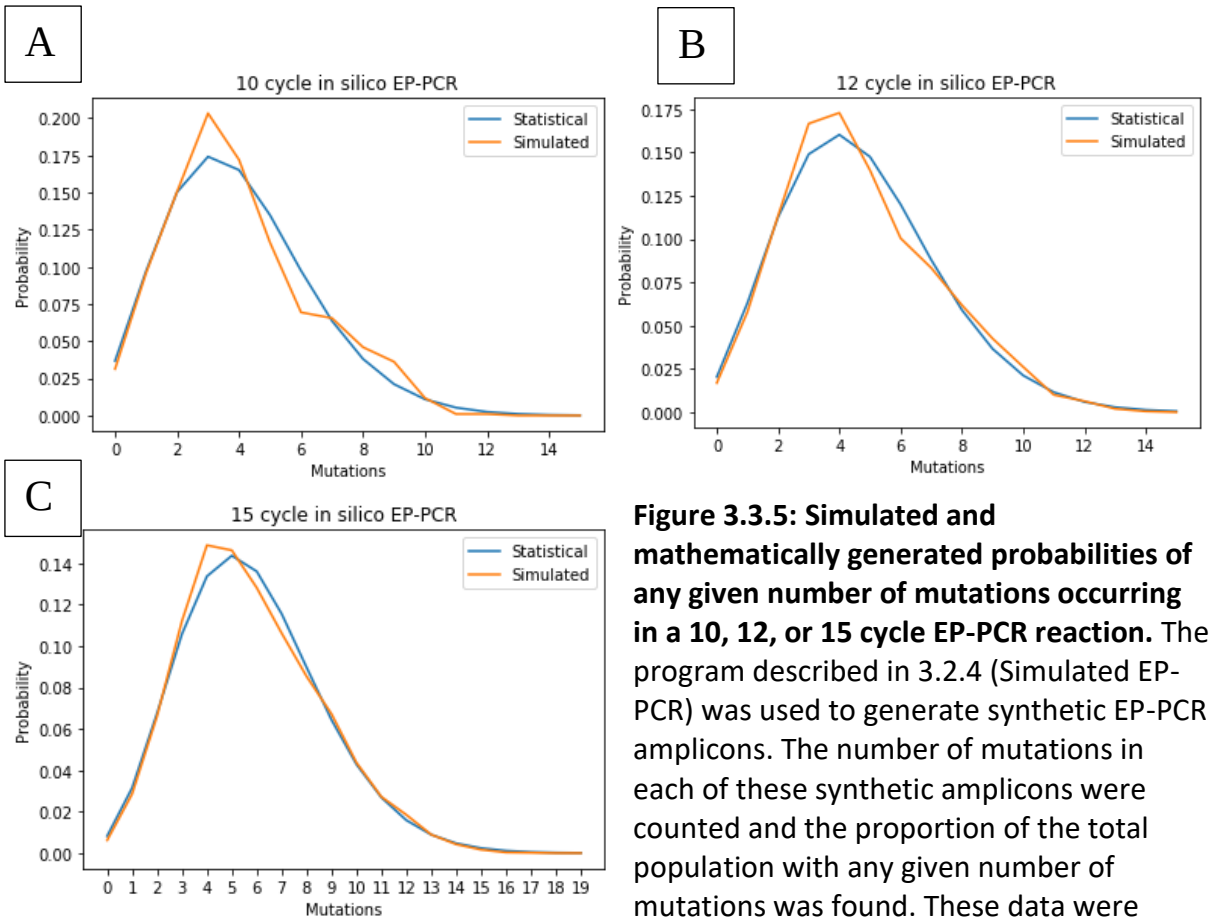


Figure 3.3.5: Simulated and mathematically generated probabilities of any given number of mutations occurring in a 10, 12, or 15 cycle EP-PCR reaction. The program described in 3.2.4 (Simulated EP-PCR) was used to generate synthetic EP-PCR amplicons. The number of mutations in each of these synthetic amplicons were counted and the proportion of the total population with any given number of mutations was found. These data were plotted as "Simulated" data for 10, 12, and 15 cycles of EP-PCR. Additionally, the mathematical modelling of EP-PCR described in 3.3 was used to predict the probability mass function of analogous EP-PCR reactions, utilising the same parameters (mutation frequency =0.7/kb; length of amplicon=1041bp; cycles=10, 12, or 15). The resultant PMF was plotted on the same graphs as "Statistical" data.

While these data show that the statistical model describes the simulated data well, there are some limitations to both systems. Notably, both the statistical system and the simulations only allow for a constant, predetermined amplification efficiency and polymerase fidelity. This is likely not accurate to reality; studies have previously shown that the amplification efficiency decreases as the PCR proceeds (Peirson et al., 2003), and that varying concentrations of dNTPs can result in incorrect nucleotide incorporation (Fromant et al., 1995). It is therefore likely that, in reality, there is a higher proportion of amplicons with fewer mutations, as the amplification efficiency decreasing would result in a higher proportion of amplicons made in earlier PCR cycles.

In this thesis, the model is also limited by the fact that a fixed fidelity of the polymerase must be known. The fidelity of polymerases is modulated by many factors, including concentration of divalent ions (Vashishtha and Konigsberg, 2016), and nucleic acid composition of the target (Pritchard et al., 2005). Specifically in this thesis, the low fidelity polymerase used (PhoEP) does not have a known associated fidelity. However, by defining the experimental conditions precisely, it should be possible to generate good estimates of the frequency of mutations introduced in an EP-PCR catalysed by the PhoEP polymerase.

3.4 Discussion

The determination of a model to predict the number and distribution of mutations that would be introduced over the course of an EP-PCR is a significant and notable development. While this work does not provide a tool to precisely predict the number of mutations that will be introduced, it can does allow parameters to be changed in order to maximise the probability of obtaining the desired number of mutations.

The relative simplicity and accuracy of the model outlined in this chapter highlights the logic behind the model. The concept of generation of DNA sequences has been presented in the literature multiple times (Moore and Maranas, 2000; Sun, 1995; Wang et al., 2000), as well as the idea that this generation distribution is affected by the amplification efficiency of the EP-PCR system (Pritchard et al., 2005). These previous models often utilise a per cycle mutation matrix, rather than the per kilobase mutation rate used in the model presented in this thesis. This mutation matrix approach likely increases accuracy of the model for specific targets, however, risks over-specificity, even within an individual EP-PCR reaction. As Moore and Maranas identified, the probability that any nucleotide *i* will be mutated to nucleotide *j* by the end of *n* extension steps depends on a per cycle mutation matrix, and if *i* had previously been mutated to nucleotide *k* (Moore and Maranas, 2000). This specificity does allow for a per cycle mutation rate, but massively increases the complexity of the model. By utilising a single, per kb extended mutation rate that is an average of all possible mutations that could occur, the possibility of mutations $i > k > j$ does not require a matrix of permutations to calculate, but instead follows basic statistics. Added specificity for the mutation rate for the model presented in this thesis could come from the mutation rate changing based on GC content of the target sequence – a concept that has been simulated previously in the literature (Pritchard et al., 2005).

There are a few limitations of the program in its current form. The most significant of these limitations is the fact that the amplification efficiency is not variable over time. This is a large departure from experimental PCRs as is evident by innumerable qPCR experiments (Peirson et al., 2003). The fact that the amplification efficiency is static will have several theoretical effects of the outcome of the experiment. Firstly, the simulated data will theoretically have a larger proportion of high number mutations than the experimental

data. This is because the amplification efficiency generally decreases as the PCR proceeds, meaning that there will be a proportion of DNA strands generated at the start of the PCR compared to the end of the PCR. As a result, the average generation number of the experimental PCR would be lower than that of the simulated PCR, which would result in a lower average length of genetic information extended. As the length of genetic information extended is proportional to the number of mutations introduced, this would mean that the simulated data would have a larger proportion of DNA sequences with higher numbers of mutations than the experimental data.

Combining the ability to maximise the probability of attaining a certain modal number of mutations, with the concept of entropy and error catastrophe introduced in Chapter 1 could result in some very powerful mutagenic tools. An error catastrophe was defined by Eigen and Schuster (1977) and is conceptually the maximum that an organism can be mutated per life cycle. The fitness of an organism increases as the mutation rate of said organism increases until the error catastrophe threshold is exceeded. At this point, the fitness of the organism rapidly decreases with increasing mutation rate. It is plausible to expand this concept to individual genes, with some key differences. While the error catastrophe of an organism is likely due to the break-down of the systems biological interactions – that is the interactions between different components of a biological system – an error catastrophe in an individual protein would not be limited by these constrictions. Instead, an error catastrophe in a single protein could theoretically occur when the primary structure of the protein becomes too distal in Hamming space to facilitate the wild-type functioning of the protein.

Owing to the redundancy of the genetic code, all genes have a certain level of intrinsic mutability before any nucleic acid change results in a change in the amino acid sequence of the protein. Expanding a level above this, due to the informatic clustering of the genetic code, wherein amino acids that share physical properties tend to be close together in Hamming space (Lenstra, 2015), even where a mutation results in a change to the primary structure of a protein, there is a likelihood that the mutation will result in an analogous amino acid substitution. This would mean that there is a probability that a change to the primary structure of the protein will have minimal effect on the secondary or tertiary structure of the protein.

4. Validation of the probabilistic model of EP-PCR

Abstract

While the work presented in chapter 3 indicated a logical and optimisable system for the introduction of mutations during EP-PCR, it is necessary to validate the models produced in said chapter. To this end, multiple EP-PCRs and subsequent sequencing reactions were carried out to find the number of mutations that are introduced by the EP-PCR while varying various parameters in the EP-PCR. Regression analysis was then carried out on these sequencing results in order to fit the sequence data to the statistical model generated in chapter 3. The result of the analysis of the sequences showed that the number of mutations introduced by an EP-PCR is variable, but generally follows a normal distribution. Furthermore, the fitting analyses facilitated the approximation of both the mutation rate of the PhoEP, and the amplification efficiency of the EP-PCR reactions. The data generated in this chapter facilitate the examination of the sequence of a mutant RTase library prior to expression and functional assay in the Chapter 5.

4.1 Introduction

Rubredoxins are a family of iron-binding proteins originally isolated from *Clostridium pasteurianum* by Lovenberg and Sobel (1965), where it was theorised that rubredoxins are a class of electron transfer proteins distinct from other known ferredoxins. It was also found that rubredoxin can carry out many of the same reactions as ferredoxins, and that rubredoxin likely contains 1 mole of iron per mole of protein.

The gene encoding rubredoxin from *Thermotoga maritima* has previously been subcloned into a plasmid vector fused to a "p53" gene from *Homo sapiens*. While it is unknown exactly what this p53 protein does, it is known to contain many histidine and proline amino acids. The abundance of these residues facilitates its tight binding to divalent metal ions, such as Ni^{2+}, or Co^{2+}. These ions are used frequently in IMAC columns, and thus facilitate purification of the Rubredoxin-p53 fusion protein.

While in chapter 3, the statistical model and the simulated data validated one another, there could be additional parameters which have not been considered in either case influencing the number of mutations introduced over the course of EP-PCR. Additionally, there are known parameters that cannot currently be incorporated into the statistical model

– such as variable amplification efficiency. The latter would reduce the value of the models as a method of predicting mutation rate, but by how much is unclear. The validation work in this chapter was devised in order to clarify this issue. In order to better validate the statistical and simulation models explored in the previous subchapter, EP-PCRs were carried out and the resultant amplicons were subsequently sequenced. Multiple templates were used in the EP-PCR - namely RTase, polyethylene terephthalate hydrolase (PETase), and a rubredoxin-p53 fusion protein. This fusion protein binds to both iron and nickel, which has been abbreviated to NiFe. The NiFe gene was used as a model to validate the EP-PCR models partly due to the relatively small size of the gene, at 456bp in length; originally, samples were sequenced by Sanger sequencing at the Core Genomics Facility at the University of Sheffield, which provided relatively short read lengths, at up to 1000bp. By using the short NiFe gene, even less successful, shorter sequencing reads could be utilised, provided that the whole NiFe protein was covered. This was especially important as due to the relatively high number of sequencing errors present in the data received from the Core Genomics Facility, a sequencing depth of at least 2 different runs was necessary to ensure that any variations from the wild type were due to mutations in the sequence, and not due to errors in the sequencing.

Later in the experimental course high-efficiency Oxford Nanopore sequencing was developed. The development of this technique conferred a higher confidence to the sequencing of nucleotides, as well as allowing longer DNA fragments to be sequenced. As such, later experiments focussed on the mutation and sequencing of a longer gene – PETase, originally from *Ideonella sakaiensis*. This allowed comparison of the frequency of mutations introduced in a longer gene.

4.2 Methods

In order to validate the statistical models and computational simulation work carried out in chapter 3, EP-PCRs were performed, and the resultant amplicon library was sequenced. A test gene – NiFe – was used for this mutagenesis experiment, followed by the gene encoding RTase, and finally a microbial PETase to in order to validate the models. The low-fidelity polymerase -PhoEP – was used to introduce mutations over the course of the PCR. This PhoEP polymerase was required to first be expressed and extracted in E. coli.

4.2.1 Expression and extraction of error-prone *P.ho* polymerase

To carry out these EP-PCRs required for validation of the statistical model, it was necessary to express and clone the error prone polymerase from *P. horikoshii* (PhoEP). As noted in chapter 3.1, PhoEP is a polymerase that introduces an increased number of mutations due to two mutations in the *P. horikoshii* replicative DNA polymerase gene. This mutant PhoEP gene had previously been subcloned into an appropriate plasmid vector and used to transform Rosetta strain *E. coli* bacteria (see Material and Methods, Table 2.1.3.1). Rosetta is a strain of *E. coli* derived from BL21(DE3), which contains a plasmid which encodes additional tRNA genes that are not present at high abundance in *E. coli*, these additional tRNAs are known to enhance expression of heterologous genes.

Rosetta pQIS257 - the strain of *E. coli* containing a plasmid vector encoding the PhoEP gene - was retrieved from a glycerol stock and plated on LB agar plates containing ampicillin and chloramphenicol. These antibiotics ensured that the strain retained both the pQIS257 plasmid, and the Rosetta plasmid respectively. After incubation overnight at 37℃, individual colonies were selected and inoculated into LB supplemented with ampicillin and chloramphenicol. These primary cultures were incubated at 37℃ overnight with shaking, before 250µl of the primary cultures were inoculated into 25ml LB Cam^+ Amp^+. This secondary culture was then incubated at 37℃ with shaking until the OD_{600} exceeded 0.5. Expression of PhoEP was then induced by the addition of IPTG at a final concentration of 1mM. The induced cultures were then returned to 37℃ and incubated with shaking for a further 4 hours, after which, the culture was transferred to a Falcon tube and was centrifuged at 5445 x g for 10 minutes at 4˚C. The supernatant was discarded, and the

resultant cell pellet was weighed, and subsequently resuspended in 5ml Taq storage buffer per 1g of cell pellet. The cell lysate was incubated at 4℃ with rotation for 60 minutes, before being sonicated at 40% amplitude for 3 bursts of 10 seconds, with 10 second break between each burst. This cell lysate was then centrifuged at 17000 x g for 10 minutes, and the supernatant was extracted and stored on ice. The cell pellet and 20µl supernatant was put aside for analysis on SDS-PAGE. The clarified cell lysate was then heated at 100 ℃ for 10 minutes and was subsequently centrifuged at 17000 x g for a further 10 minutes. The supernatant was extracted and used in PCR experiments as the DNA polymerase. The pellet and 20µl supernatant were put aside for analysis on SDS-PAGE.

The supernatant and pellet from various points during the PhoEP extraction process were analysed on an SDS-PAGE (see figure 4.2.1) to verify that a band corresponding to the PhoEP protein was present. Additionally, the extraction process was repeated with uninduced cultures, to show that the corresponding band was only present when induced by IPTG.

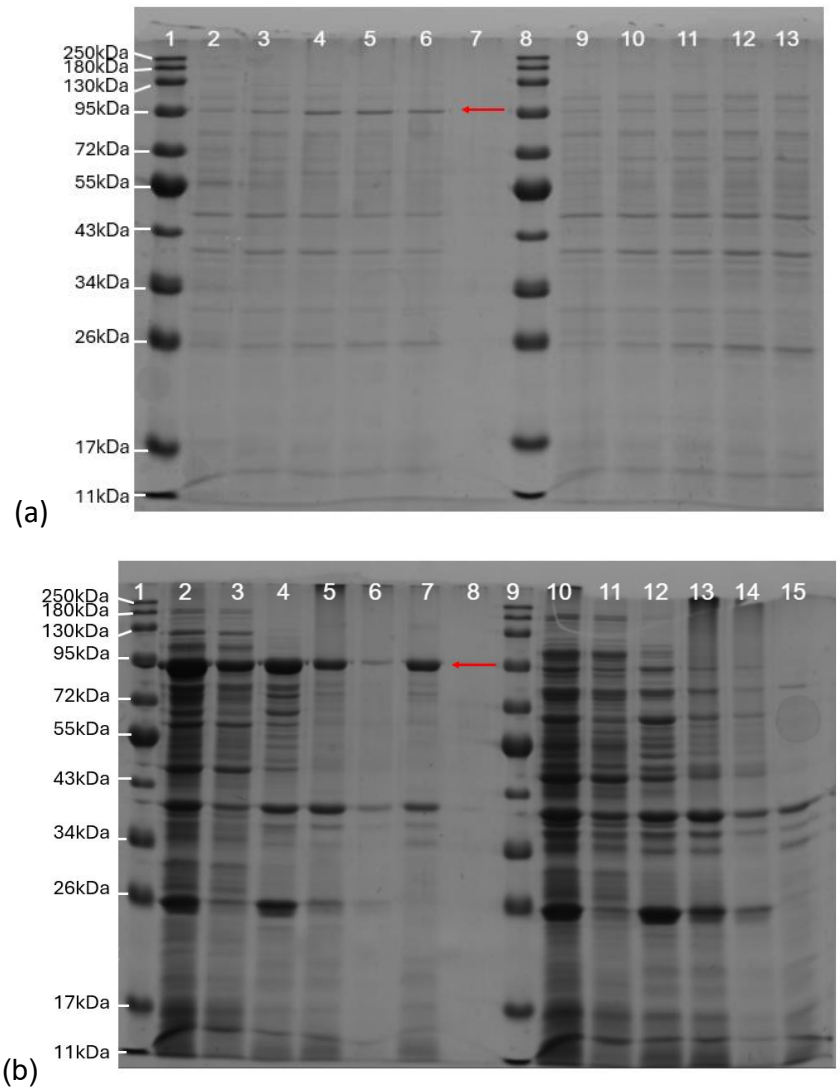


Figure 4.2.1 – SDS_PAGE analysis of the various stages in expression and enrichment of recombinant PhoEP in *E. coli (*Rosetta*)*.
(a) PhoEP was expressed and purified as described in 4.2.1. **Lanes 2-6** show the protein extracted from the induced culture after 0-4 hours respectively. Similarly, **lanes 9-13** show the protein extract from the uninduced culture after 0-4 hours respectively. A clear band at approximately 95 kDa is present in the induced sample, but not in the uninduced sample. This band has the expected electrophoretic motility as the PhoEP protein. **(b)** Expressed PhoEP was extracted and enriched following the protocol in 4.2.1. **Lanes 2-7** show the protein extracted from the induced culture, whereas **lanes 10-15** show the protein extracted from the uninduced culture. From left to right in each group, the lanes show: **2 and 10**-sonicated culture; **3 and 11 -** pellet from sonicated culture; **4 and 12 -** supernatant from sonicated culture; **5 and 13** - heated culture; **6 and 14** - pellet from heated culture; **7 and 15** - supernatant from heated culture. It is evident from lanes 2-7 that the heating of the lysed cells aids in the purification - comparing lanes 4 and 7 shows that the band at approximately 95 kDa is better purified in the soluble fraction of the heated sample (lane 7) than in the unheated sample (lane 4). Additionally, the uninduced samples (**lanes 10-15**) show no significant band at 95kDa before or after heating.

4.2.2 Optimisation of PhoEP concentration in EP-PCR

Prior to utilisation of the extracted PhoEP in EP-PCR, the experimental conditions for using PhoEP in a PCR required optimisation. It was found early on that the concentration of polymerase added to the EP-PCR played a vital role in the success and yield of PCR product. To this end, a series of experiments were conducted to ascertain the optimal volume of PhoEP preparation to add to the PCR in order to maximise expected product yield.
Newly isolated PhoEP was diluted in Taq storage buffer (see Chapter 2.1.2), and different volumes of various dilutions were used in a 25 µl PCR reactions. The PCR products were then loaded onto an agarose gel (see figure 4.2.2(A)). Both the NiFe gene and the RTase gene were used as templates DNA for these PCRs.

It was found that the addition of undiluted PhoEP resulted in low molecular weight nucleic acid "smears" on the gel (see figure 4.2.2(A), lanes 5 and 6). This might be due to carry over of inhibitors or other contaminants from the cell lysis. A 10-fold dilution of the enzyme preparation yielded satisfactory target amplification, as shown by a single band present in figure 4.2.2(A) lanes 7 and 8. A further 10-fold dilution yielded no visible band on the agarose gel, suggesting that the polymerase concentration is too low to result in a detectable PCR product.
A further experiment was conducted in order to fully elucidate the amount of PhoEP that should be added to a PCR reaction for optimal results. To this end, a series of 50 µl PCRs were set up with varying volumes of PhoEP added. 1-10 µl of PhoEP diluted 10-fold in Taq storage buffer was utilised in this experiment, with the pJF012 plasmid acting as the template DNA. T7 forward and reverse primers were used to amplify the RTase gene and produce a PCR product with an expected size of 2456bp. It was found that all volumes of PhoEP added above 1 µl, yielded a band on the agarose gel of the expected molecular weight (see figure 4.2.2(B)). There was a slight increase in the band intensity from the addition of 2 µl to 4 µl PhoEP added (see figure 4.2.2(B) lanes 3-5), which then plateaued with larger volumes of PhoEP. From this experiment, it was decided that the optimal volume of PhoEP to add was 4 µl of a 10-fold diluted sample.

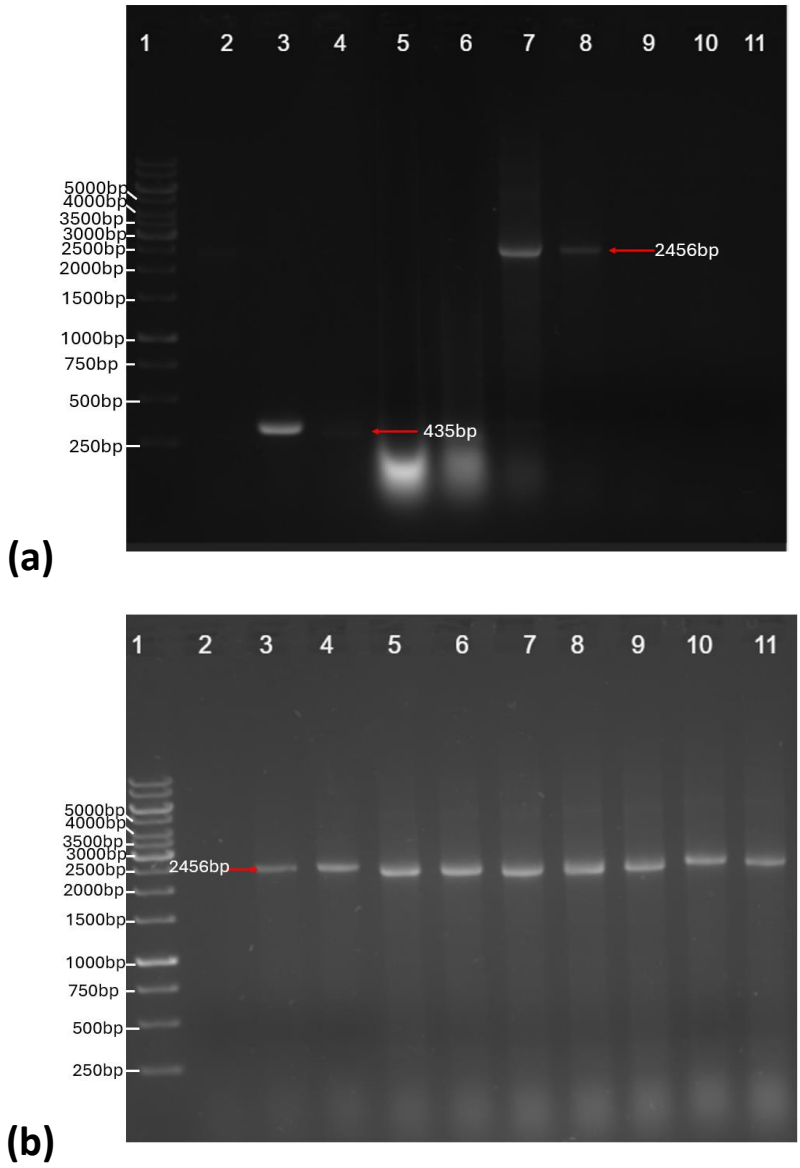


Figure 4.2.2 – Analysis of the outcome of PhoEP catalysed amplifications carried out to optimise the composition of PCR reactions. (A) PCR reactions were carried out in 25µl volumes to compare batches of PhoEP. T7F2 and T7R2 primers were used on a pJF012 template (expected length 2400bp) in all but lanes 3 and 4, where a pQIS207 template was used (expected length 500bp). The older polymerase stock was used in **lanes 2-4**, and varying dilutions of the new PhoEP extraction were used in **lanes 5-11**. The PCR reactions loaded into lanes 5 and 6 contained 5µl and 1µl undiluted PhoEP extract respectively, **lanes 7 and 8** contained 5µl and 1µl PhoEP diluted 10 fold in Taq storage buffer, and **lanes 9 and 10** contained 5µl and 1µl PhoEP diluted 100 fold in Taq Storage Buffer. A PCR reaction containing no polymerase was loaded into lane 11. All PCR reactions were run using the "60An_2mEx" protocol (see appendix). **(B)** A follow-up PCR experiment to find the optimal volume of PhoEP extract to add to PCRs was carried out using a larger reaction volume of 50µl. The new PhoEP stock was diluted 10-fold in Taq Storage Buffer, and volumes in increasing increments of 1µl were added to the PCR. The PCR reaction loaded into lane 2 contained 1µl of 10 fold dilute PhoEP, while the PCR reaction loaded into lane 11 contained 10µl of 10 fold dilute PhoEP. T7F2 and T7R2 primers were used on a pJF012 template, giving an expected length of ~2456bp. The PCR was run using the "60An_2mEx" protocol (see appendix). From these results, it was decided that 4µl PhoEP extract should be used in 50µl PCRs, or 2µl in 25µl reactions.

4.2.3 Random mutagenesis and sequencing of the NiFe gene

The gene encoding the NiFe protein was randomly mutated by amplifying the pQIS207 plasmid using PhoEP. Rubredoxin Forward and Rubredoxin Reverse primers (see appendix 1) were used to amplify the NiFe gene from pQIS207. Different numbers of cycles were used to amplify the NiFe gene, ranging from 10 to 25. Each EP-PCR run was kept separate. The resultant random amplicon libraries were purified using an ISOLATE II PCR clean-up kit (Bioline) and subcloned into a pUC19 plasmid.

This library was then used to transform NEB5α strain *E. coli*, which were plated on LB agar plates, supplemented with ampicillin. Transformants were replated and the presence of NiFe coding sequences established using colony PCR. NiFe recombinants were then used for plasmid purification followed by nucleotide sequencing at the Core Genomics facility. In total, 76 NiFe clones were sequenced using this method.

4.2.4 Random mutagenesis and sequencing of reverse transcriptase

The open reading frame encoding MMLV RTase was synthesised by Eurofins Genomics and was subsequently subcloned into an expression plasmid (see chapter 5.2 for details). This plasmid containing RTase (pJF012) was then used as the template in an EP-PCR, utilising the PhoEP. T7F2 and T7R2 primers (see appendix 1) were used to amplify the RTase gene, yielding a linear DNA product of 2456bp in length (see figure 4.2.2(B)).

The resultant RTase random amplicon library was purified using a Bioline ISOLATE II kit, before being subcloned into a pET28a vector. The ligation product was then used to transform NEB5α competent cells, which were plated onto LB Kan^+ agar plates. After growing overnight at 37°C, individual colonies were picked and streaked onto new LB Kan^+ agar plates. From these streak plates, primary cultures were made, which were subsequently used to extract plasmids, and those containing the RTase gene were sequenced at Eurofins Genomics, utilising Sanger sequencing with four primers to ensure sequence coverage. In total 95 clones were sequenced using this method.

4.2.5 Random mutagenesis and sequencing of a polyethylene terephthalate hydrolase

In order to elucidate any concerns with the previous sequencing carried out over the course of this work, a further series of random mutagenesis and sequencing experiments were carried out utilising a plasmid encoding a PETase as the target. This gene has been the subject of extensive directed evolution attempts with some positive results in the form of improved enzyme thermostability (Bell et al., 2022), and higher substrate turnover characteristics (Tournier et al., 2020). These breakthroughs represent evidence that in some cases, directed evolution has the potential to yield beneficial results in relatively short timescales.

The gene for PETase was obtained from TWIST Bioscience (USA). A plasmid encoding the PETase gene in a pET28a vector incorporating a segment of DNA encoding the NiFe protein as an N-terminal fusion to the PETase, was used in these experiments. The addition of the NiFe sequence was made to facilitate purification of PETase using Ni-NTA chromatography. The NiFe tag also confers a red colouration to PETase fusion, allowing easy tracking throughout its purification.

As the NiFe-PETase fusion was under control of a T7 promoter, it was possible to use this construct directly in an EP-PCR utilising the T7 forward and reverse promoters. Two different random mutagenesis experiments were carried out on this NiFe-PETase construct; initially, PhoEP was used to randomly mutate the gene, followed by a more traditional EP-PCR protocol involving Taq polymerase with Mn^{2+} as the divalent cation in place of Mg^{2+}. This allowed comparison of the number and type of mutations produced by each method. The total length of NiFe-PETase gene that was extended was 1593bp, with 1347bp of this representing the wild-type open reading frame (ORF) of the gene.

4.2.5.1 Random mutagenesis of PETase using PhoEP

PhoEP was initially used to randomly mutate the PETase gene. 10ng pPETase was used as the template in a 50 µl EP-PCR reaction using PhoEP as the polymerase. T7 forward and reverse primers were used to specifically amplify the NiFe-PETase fusion gene. The PCR was run for 35 cycles with 30s denaturing at 95 °C, 30 seconds annealing at 62 °C, and 120

seconds extension at 72 °C. This cycle was preceded by an initial denaturation at 95 °C for 5 minutes and proceeded by a final extension at 72 °C for 10 minutes.

After the EP-PCR, the reaction products were loaded onto an agarose gel electrophoresis. A 1593bp band was excised from the gel and purified using an ISOLATE II PCR and gel kit (Bioline). The resultant DNA was eluted into 40 µl ddH_2O giving a concentration of 6.9 ng/µl and was digested with XbaI and XhoI (NEB), and ligated into a pET28a vector using T4 DNA ligase (NEB). *E. coli* BL21(DE3) competent cells were finally transformed using these ligation reactions and were plated onto LB agar containing 50 µg/µl kanamycin.

Individual colonies from the resultant transformation plates containing the NiFe-PETase plasmids were verified by colony PCR. Once confirmed, the clones were replated in a regular grid onto LB agar plates containing 50 µg/µl kanamycin for subsequent nucleotide sequencing.

4.2.5.2 Random mutagenesis of PETase using Taq/Mn^{2+}

In addition to the random mutagenesis using PhoEP, the NiFe-PETase gene was also randomly mutated utilising an EP-PCR containing Taq polymerase and 0.32mM $MnCl_2$ in place of $MgCl_2$. T7 forward and reverse primers were used to selectively amplify the gene NiFe-PETase gene from the plasmid construct. 1ng plasmid was used in the reaction. The same PCR program was used for this EP-PCR as for the mutagenesis using PhoEP.

After the PCR, 50 µl of product was purified using an ISOLATE II PCR and gel kit, and eluted in 40 µl, at a final concentration of 38.2 ng/ µl. This DNA product was digested with EcoRI and NotI (NEB) prior to ligation into a pET28a vector. Additionally, the DNA product was digested with DpnI (NEB) to remove any residual template DNA. The DNA product was ligated into pET28a, and these ligation products were subsequently used to transform BL21(DE3) strain *E. coli.* The presence of the NiFe-PETase gene-containing plasmids was verified by colony PCR, and positive clones were replated in a regular grid onto LB agar plates containing 50 µg/µl kanamycin.

4.2.5.3 Oxford Nanopore sequencing of NiFe-PETase clones

NiFe-PETase recombinant colonies were sequenced using an Oxford Nanopore MinION device. In order to facilitate appropriate sequencing depth of the individual primers, a library preparation protocol was developed. In this protocol, 192 unique primer barcodes were designed (96 forward; 96 reverse) (See appendix 1.2). These barcodes contained a primer sequence allowing binding to the T7 promoter or terminator, with a 5' unique barcode. These barcodes were designed such that there would be maximal hamming distance between them in order to minimise possibility that a sequencing error would result in one barcode being mistaken for another.

Each of the colonies to be sequenced were inoculated into a PCR reaction, utilising one of the barcoded primer pairs, and a high-fidelity PCR mastermix (NEBNext High-Fidelity 2X mastermix; NEB). The PCR products were then combined (12 PCR products were mixed together into a single tube) and purified using an ISOLATE II PCR and gel kit (Bioline). The concentration of each of these pools were then measured using a Nanodrop spectrophotometer, the pools of 12 were then combined into 96 such that each pool of 12 was normalised to a concentration of 10 ng/µl.

The 96 pooled samples were then "end-prepped" using FFPE DNA repair enzyme and Ultra II End prep enzymes (NEB), and Nanopore sequencing adapters were added. The sample was then loaded onto an Oxford Nanopore flow cell or Flongle which was used to sequence the 96 pool.

The resultant sequencing data were "basecalled" by the MinKnow software from Oxford Nanopore. This procedure generated fastq data files which were then demultiplexed into their respective barcodes using MiniBar software (Krehenwinkel et al., 2019). After demultiplexing, Canu – an open-source assembly program - was used to assemble each barcoded sample into a coherent amplicon with sufficient read depth (Koren et al., 2017). A shell script was written in order to perform these programs (see appendix 2.11).

The sequences were then analysed in comparison to the wild-type NiFe-PETase gene, and the number and type of mutations were found. In total 208 PhoEP mutated NiFe-PETase clones, and 222 Taq/Mn^{2+} mutated NiFe-PETase clones were sequenced and analysed in this manner.

4.2.6 Construction of EP-PCR models

Models outlined in section 1.5 were replicated using equations shared in the respective papers. To this end, the models from Moore and Maranas (2000) (known as the Moore model), Wang et al., (2000) (known as the Wang model), and Pritchard et al., (2005) (known as the Pritchard model) were replicated in Python programming language. The models were constructed within a function such that parameters could be easily varied, and to facilitate non-linear least squares fitting of the parameters to the sequence data. The program written to call these models can be found in appendix 2.09.

4.3 Results

4.3.1 NiFe sequencing results

The nucleotide sequences of the NiFe plasmids were obtained using 2 different sequencing primers – ((LacZα F and LacZα R) see appendix 1). The output of the mutation and sequencing pipeline were 152 sequences, corresponding to the forward and reverse strands of the mutant genes. As such, this represents 76 unique sequences. Of these 76 sequences: 32 are sequences from 10 cycles of EP-PCR; 23 are sequences for 20 cycles of EP-PCR; and 21 sequences are from 25 cycles of EP-PCR. These sequences had an average length of 1072bp, providing ample read length for the 435bp amplicon, even with the 521bp *lacZa* gene fragment either side of the amplicon. Some of this mutation and sequencing work was carried out by Alex Wakeman – a Masters student in the laboratory.

These sequencing results were then analysed to identify any mutations in both the forward and reverse sequences in order to ensure that the apparent mutations were not a result of sequencing errors. Any sequences with discrepancies between the forward and reverse strands were removed from the dataset. The resultant confirmed mutations were then compared to the wild-type NiFe sequence, and the number and type of mutations were recorded (see figure 4.3.1).

The resultant data revealed there had been variation in the number of mutations introduced at each cycle number of the EP-PCR tested (see figure 4.3.1(A)). This is consistent with the predictions made by the EP-PCR models outlined in chapter 3, wherein it was predicted that each EP-PCR would generate a wide range of mutations.

There are some concerns with the data collected. For example, there are several abnormal bases present in the dataset – such as W, Y or S. These sequence results indicate that there might be several different bases at that point in the sequence. There are a number of possible explanations for these anomalies: the plasmid supplied for sequencing could have been a mixture of two different mutant plasmids, or the sequencing reaction could have failed due to a suboptimal concentration of plasmid.

Additionally, the standard deviation and mean of the results do not follow the expected trajectory as number of cycles increases; the model predicts that as the number of cycle increases, both the average number of mutations, and the standard deviation of the number of mutations would increase. This is not observed in the sequencing data, where

the average number of mutations per clone decreased from 1.28 in 10 cycles, to 1.08 mutations per clone in 20 and 25 cycles. Furthermore, the standard deviation of the sequenced mutants varied from 2.44 mutations per clone in 10 cycles, to 2.81 mutations per clone in 20 cycles and 1.66 mutations per clone in 25 cycles. This discrepancy between the modelled data and the sequenced data could have several explanations, however the most likely one is that the model is inadequate. This could be due to a missing parameter, or it could simply have faulty logic. One potential missing parameter is the variable amplification efficiency observed in qPCR experiments, which can have a marked effect on the distribution of mutations in the EP-PCR.

A different explanation for the differences between the predicted outcome and the actual outcome is the low number of mutants sequenced and successfully analysed. 28, 23, and 12 sequences were sequenced for 10, 20 and 25 cycles respectively. These are low numbers compared to the total number of amplicons generated in an EP-PCR. Furthermore, the sequences were not generated in a single EP-PCR experiment; the amplicons that were sequenced were generated in numerous different EP-PCR experiments, both between cycles and sequences of the same number of cycles. From the work described in chapter 3.2.3, it is evident that, in the simulated EP-PCR model, the mean number of mutations introduced in an EP-PCR will vary, even when all parameters are kept constant. This means that there could have been large variation within each sample, and that the sequenced data might not be the product of a single dataset.

In order to improve these data, it would be beneficial to increase the quantity of the sequencing carried out on every amplicon library. To do so would require more time focused on this issue, or a better experimental protocol, allowing a higher throughput of mutants and sequencing. This improved protocol would use next generation sequencing (NGS) techniques, such as Pacific Bioscience sequencing, or Oxford Nanopore sequencing. Additionally, the data would be improved by having distinct separate EP-PCRs with repeats for different cycle numbers kept separate. This would allow examination between EP-PCRs of the same parameters, thereby allowing validation or rejection of the findings in chapter 3.2.3.

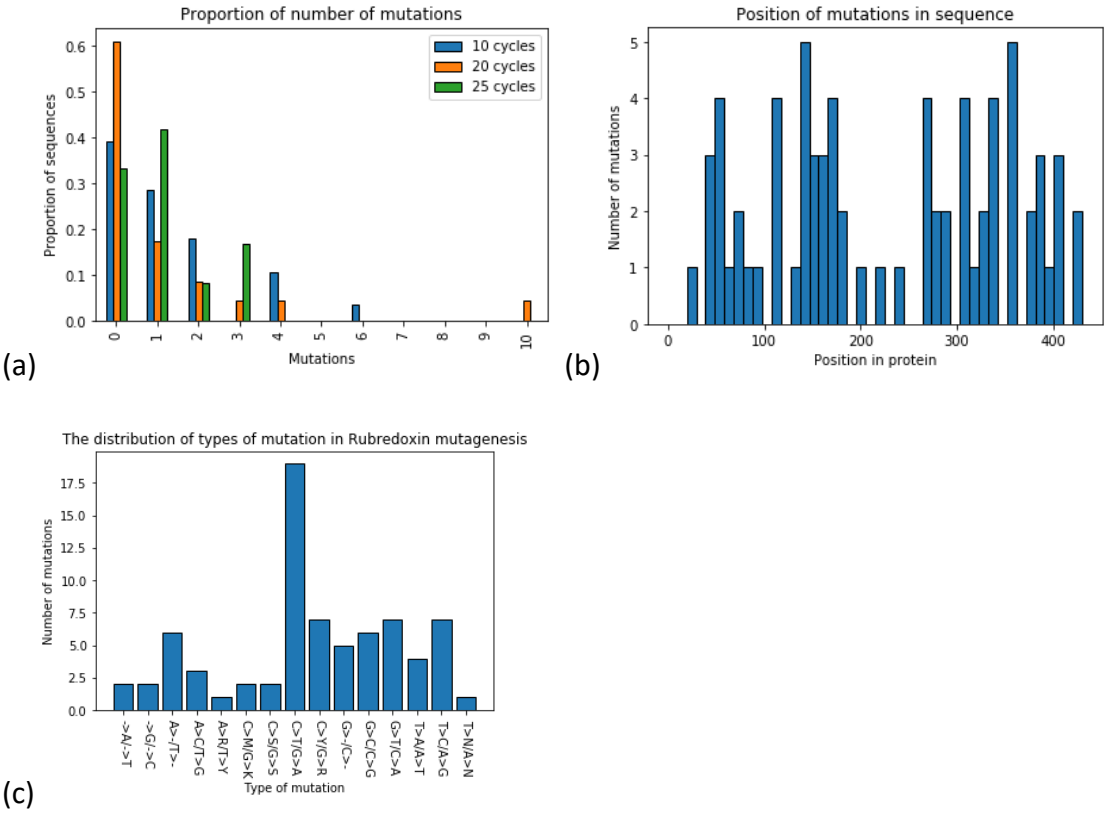


Figure 4.3.1: Various graphs depicting analyses of the sequencing results of p53-Rubredoxin. (a) The proportion of amplicons with any number of mutations was plotted on a bar chart. Mutations were identified by performing a Needleman-Wunch alignment between the wildtype NiFe-rubredoxin protein, and the resultant number of mutations per amplicon were saved in a list. Once all sequences had been analysed in this manner, the resultant numbers were plotted on a bar chart. **(b)** The position of each mutation in the sequence was found and plotted on a histogram. While there are no extensive mutation hotspots, there are regions with more mutations, however, this might be due to the additive nature of mutations in EP-PCR. **(c)** The type of each mutation was found. In this graph, the type of mutation is denoted with a ">", eg. G>A would denote a mutation from guanine in the wildtype to adenine. Additionally, as it is unknown which strand the mutation occurred on, both possible mutations are represented in a single bar. Some sequencing errors occurred, wherein it is not known what exact mutation occurred, and are represented using IUPAC nomenclature of bases. From this analysis, it is clear that some mutations occur much more frequently than others. Interestingly, it seems that cytosine and guanine are more likely to be mutated than adenine and thymine, despite the wildtype sequence having only 48% CG content.

4.3.2 Nonlinear Least Squares analysis of NiFe sequences

Following on from the determination of the frequency of mutations in each reaction, nonlinear least squares analysis was used to find the theoretical mutation rate of the reaction. This utilised the mathematical model outlined in section 3.3.4 to generate a model with variable parameters, which could then be compared to the experimental data, and the resultant residuals could be minimized.

A Python program was written for this least squares analysis (see appendix 2.7 and 2.8), utilising the "scipy" module, and specifically the "optimize.least_squares" function. This function acts by taking a separate function as an argument which in turn calculates the difference between the real and simulated data for any given parameter or set of parameters. The "least_squares" function will then vary the parameter to minimise the residuals between the real and simulated data.

Initially, only the error rate of the reaction was varied in the nonlinear least squares program. The results of this nonlinear least squares analysis showed that the estimated mutation rate of the polymerase was between 7.9×10^{-5} and 1.7×10^{-4} per kilobase extended. This is lower than the mutation rate quoted by Biles and Connolly (2004), wherein a mutation rate of approximately 7×10^{-4} per kilobase extended was observed for a similar PfuEP. This discrepancy could be due to the fact that while the polymerase from *P. furiosus* and the polymerase from *P. horikoshii* are homologous, there is some difference in the amino acid as they share approximately 80% sequence identity. This means that the amino acid changes that have been replicated in *P. horikoshii* polymerase might have a slightly different effect on the introduction of mutations. Alternatively, it could be due to low number of sequenced mutants.

A further non-linear least squares analysis was subsequently carried out, varying both the mutation rate, and the amplification efficiency of the EP-PCR. The same raw data were input into this program, and a constant amplification efficiency was varied in addition to the mutation rate to minimise the residuals. The ability to vary the constant amplification efficiency generally acted to increase the optimal predicted mutation rate of the EP-PCR; the predicted mutation rate of both 10 and 20 cycles of EP-PCR increased either 6.43-fold or 28.65-fold respectively. This is due to the decreased amplification efficiencies of both reactions, at 0.094 (1) and 0.031 (1) respectively. The result of these changes on the fitting

showed a minor improvement on the fit of the statistical data to the sequenced data, as is evidenced by the decrease in the Mean Absolute Error (MAE) of the fitted data from 0.009265 (1) to 0.006639 (1) in 10 cycles, and 0.008940 (1) to 0.001585 (1) in 20 cycles. For 25 cycles, the nonlinear least squares analysis determined that the optimal value of amplification efficiency was 0.5. This is equivalent to the “perfect amplification” assumed in the previous regression, and so there is no change in the fit of the data, or the MAE.

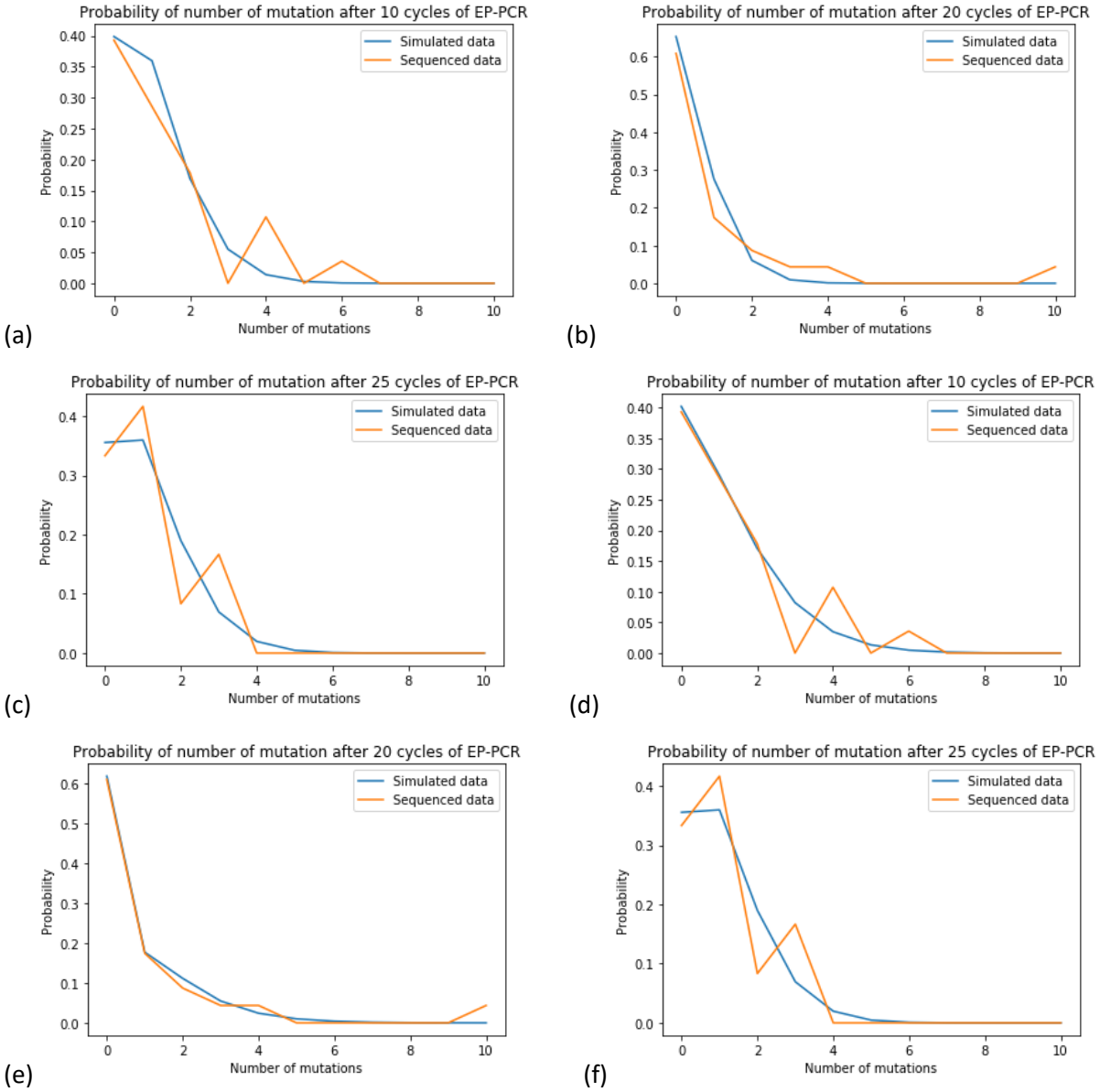


Figure 4.3.2: Non-linear least squares analysis of proportion of mutations introduced over 10, 20, or 25 cycle EP-PCR into p54-Rubredoxin. Non-linear least squares was used to vary either mutation rate (**a-c**), or mutation rate and amplification efficiency (**d-f**), to fit simulated data to the experimental data. The output parameters were then used to generate simulated data (blue) which were plotted alongside the experimental data (orange)

Number of cycles	NLLS mutation rate only		NLLS mutation rate and amp efficiency		
	Mutation rate	**L1 loss**	**Mutation rate**	**Amplification efficiency**	**L1 loss**
10	0.000171	0.00926	0.0011046	0.0935338	0.00663
20	0.0000786	0.00894	0.0022525	0.03054653	0.00158
25	0.0001933	0.01245	0.0001933	0.5	0.01245

Table 4.3.2: The optimised mutation rate, or mutation rate and amplification efficiency with associated L1 cost function following a non-linear least squares regression analysis.

4.3.3 Reverse transcriptase sequencing results

Following on from the sequencing of the NiFe random mutants, further sequencing was carried out on the RTase from MMLV. The utilisation of RTase as a secondary test gene was twofold; a main aim of this work was to increase the thermostability of RTase, meaning that the validation of the models could be carried out in parallel to the generation of mutants, and the RTase gene is around 5 times the length of the NiFe gene, allowing the exploration of the effect of gene length on the rate of introduction of mutations.

RTase mutants were sequenced at Eurofins Genomics, using four primers. These primers should have ensured that the whole length of the RTase was covered in multiple sequencing results, however, due to irregular sequencing result lengths and failed reads, some regions had single coverage, and some regions had no coverage.

Despite the regions of single coverage, the quality report submitted by the sequencing service indicated good quality reads based in most instances (Sanger QC score >30) and allowed most of the sequence reads to be used.

On average there is a 75bp gap where no sequencing was registered. As a result of this gap in sequence reads, it is impossible to completely ascertain how many mutations have been introduced into each different sequence, as any number of mutations could theoretically have occurred in the 75bp gap. However, it is assumed that the frequency of mutation in the 75bp gap is the same as all downstream mutations. Additionally, it is possible to determine the number of mutations per kilobase of sequenced genetic material. This metric could provide imprecise measurements for the error rate of the EP-PCR but should provide valid data for the determination of accuracy of the statistical models of EP-PCR.

The sequenced genes were assembled using sequence alignment software. Any discrepancies between sequence reads of the same mutant were resolved using the Sanger QC result. These assembled sequences were then analysed using various Python programs to determine the number of mutations per sequence, the types of mutations that occurred, and the number of mutations per kilobase of DNA sequenced. These Python programs counted the number of mutations from the wild-type and stored them in a database.

These analyses showed that the number of mutations introduced does follow a normal distribution, as was predicted with the statistical model of EP-PCR (chapter 3.3.4), albeit with a greater than expected number of sequences with no mutations. There are several

possible explanations for this discrepancy; the excess unmutated sequences could be leftover template DNA present in the final EP-PCR product (see figure 4.3.3C)
Alternatively, as the models assume a constant amplification efficiency of the PCR and the experimental EP-PCR is more likely to exhibit a decreasing amplification efficiency, the excess unmutated amplicons could be due to increased amplification efficiency at the start of the reaction, where the "generation" number and thus length of genetic material amplified is low, compared to the end of the reaction. This would mean that there would be increased amplification of sequences with fewer average mutations at the start of the EP-PCR, and decreased amplification of sequences with a higher number of average mutations at the end of the reaction. This hypothesis would however dictate that the normal distribution would be shifted and skewed over to the right-hand side.
Finally, the discrepancy from the model could be a sampling error. As the potential dataset of the EP-PCR experiment is very large – potentially over $1x10^6$ sequences – there is a large probability that the 95 amplicons sequenced are not fully indicative of the dataset at large. This could mean that either the unmutated sequences are over-represented, or equally that the single mutations are under-represented. While a greater number of sequences would resolve the sampling issue, it may be necessary to introduce modifications to the EP-PCR experimental design in order to minimise the attrition of the rate of amplification as the EP-PCR progresses.

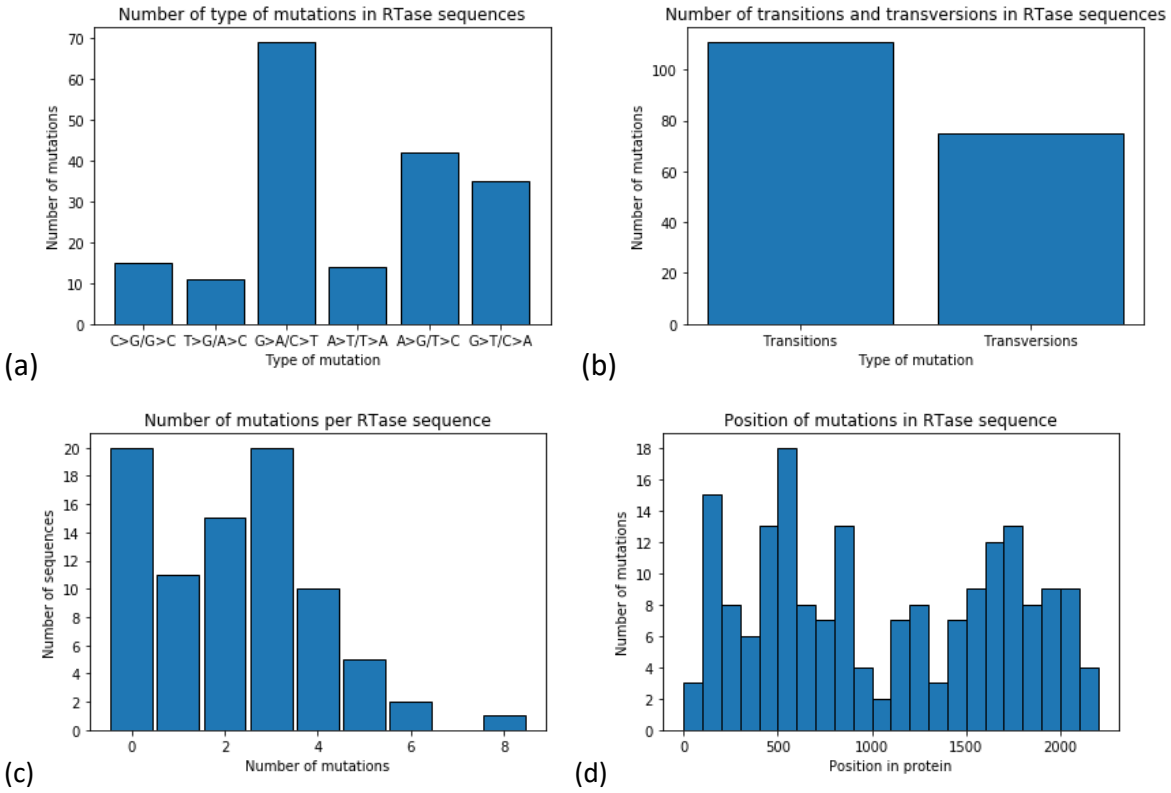


Figure 4.3.3: The mutational profile of 95 RTase clones. The sequencing results from RTase sequencing reactions were analysed, and the types of mutations were quantified. **(a)** Shows the identity of mutations in 95 RTase mutations. As mutations can be introduced in either the coding or non-coding strand of DNA, there are two different types of mutation that could occur in any reported instance of mutation. **(b)** Shows the number of transition and transversion mutations. **(c)** Depicts the number of sequences found to have each number of mutations from the original template RTase. **(d)** A histogram showing the locations of the mutations over the gene.

4.3.4 Non-linear least squares analysis of RTase sequencing data

Non-linear least squares regression was used to predict the error rate of the PhoEP. To this end, a Python program (see appendix 2.12) was written in to carry out non-linear least-squares analysis on the sequencing data and the mathematical data described in chapter 3.3.4. The details of this program are outlined in chapter 4.2.4. Initially only the error rate of the reaction was optimised, with the other parameters (gene length, amplification efficiency, and cycle number) being hardcoded into the program. Most of these parameters were known and constant for the EP-PCR in which the mutants were generated, with 2152bp gene length (of which 2077bp was sequenced), and 25 cycles of EP-PCR. The amplification efficiency was not known for the specific EP-PCR experiment that generated the mutants.

The initial non-linear least-squares analysis found that the EP-PCR had an error rate of 9.545×10^{-5} per base pair. This means that one mutation will be introduced for every 10476 bases of DNA extended. This is a decrease in mutation rate by approximately 10 fold from the data presented by Biles and Connolly (2004), who found that the error rate of the error-prone *P.fu* polymerase was 7×10^{-4}. As the PhoEP and PfuEP are fairly similar and have analogous mutations, it would be expected that they would have a similar mutation rate. The evidence from the Rubredoxin sequencing and subsequent fitting did not necessarily confirm that PhoEP and PfuEP have a similar error rate, with a large variation of mutation rates for PhoEP calculated.

As the amplification efficiency of the reaction was not known and was assumed to be ideal in the previous least-squares analysis, a secondary least-squares analysis was carried out optimising both the error rate and the amplification efficiency of the reaction.

The same Python program was used, with the amplification efficiency added to an input list alongside the error rate, such that both could be passed to the least_squares function to optimise.

This least-squares analysis found that the amplification frequency of the reaction was relatively low, at just 7.942×10^{-2}. For comparison, in this system a perfect amplification efficiency wherein each strand of DNA is doubled in each cycle would have an amplification efficiency of 0.5.

In concert with this lower amplification efficiency, the estimation of the error rate of the reaction increased when optimised alongside the amplification efficiency, ultimately giving a value of 6.239×10^{-4}, an almost 10-fold increase from the previous optimised value in the absence of amplification efficiency. This value is also much closer to the value posed by Biles and Connolly (2004).

This noted increase in error rate alongside the decrease in amplification efficiency makes logical sense; as the amplification efficiency is lower than in the previous analysis, there are fewer amplicons produced, thus a lower average generation number of the amplicons. This means that the length of genetic material extended to produce any one amplicon is also decreased, which means that a higher error rate is required to attain the same distribution of mutations over the reaction.

Utilising a differential constant amplification efficiency also resulted in a slightly better fit for the data, with the MAE decreasing from 0.0167 (1) to 0.00658 (1) when amplification efficiency was optimised alongside mutation rate.

While this latter nonlinear least-squares analysis allowed the optimisation of a static amplification efficiency, previous publications (e.g. Peirson et al., 2003) have shown that the amplification efficiency is not static over the course of the reaction, but rather starts high, and decreases as the reaction proceeds. While the static amplification efficiency provides a reasonable metric as to the average level of the amplification efficiency over the whole reaction, a mutable amplification efficiency would have some alternative, far-reaching effects on the error-profile of the amplicons. For example, an increased amplification efficiency at the start of the reaction compared to the end would mean that there would be a greater proportion of low generation amplicons, which would correspond to more amplicons with fewer mutations.

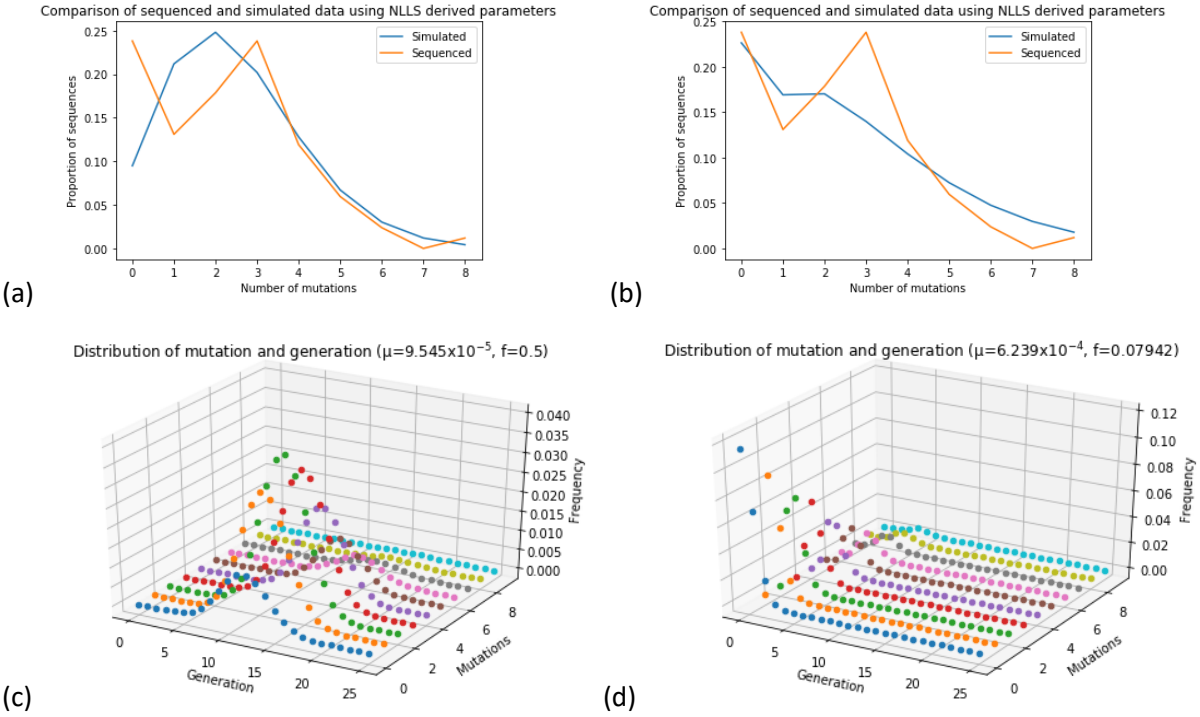
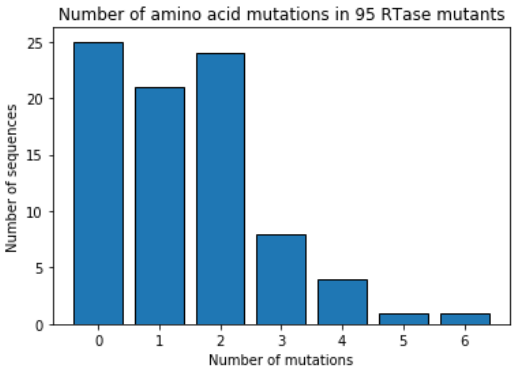


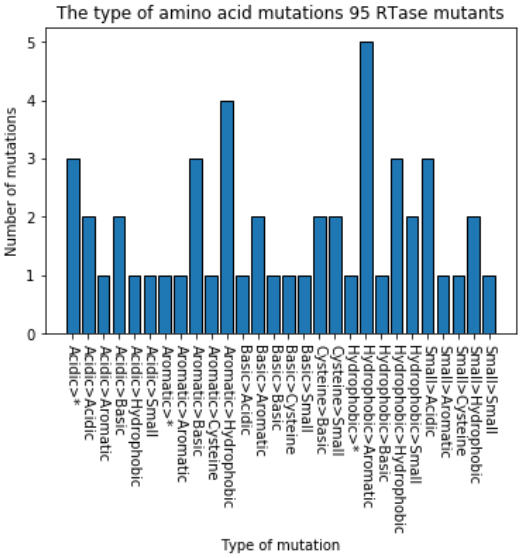
FIGURE 4.3.4: Non-Linear least squares regression with (a and c) or without (b and d) concurrent regression of amplification efficiency. Non-Linear Least-squares analysis was used to estimate the mutation rate of the PhoEP either assuming perfect amplification efficiency **(a)**, or assuming constant, imperfect amplification efficiency which was also estimated by NLLS analysis **(b)**. These parameters were then used to simulate the probability of mutations and generations of amplicons in a simulated EP-PCR based on the work carried out in chapter 3.3.4 **(c and d)**.

4.3.5 Effects of random mutagenesis on RTase protein sequence

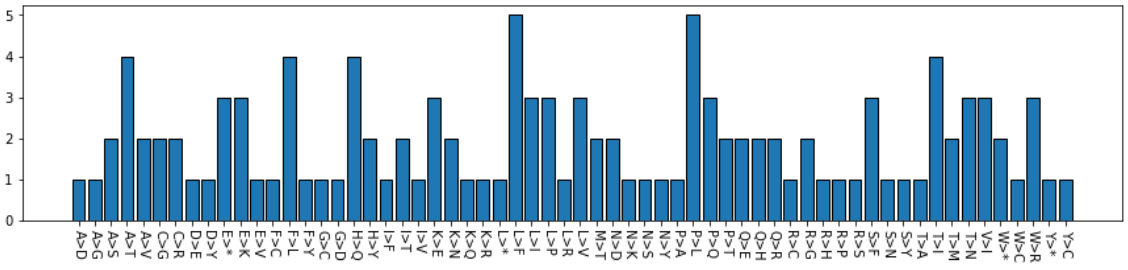
Owing to the degenerate nature of the DNA triplet code, any mutation in the DNA of a sequence is not necessarily representative of a mutation to the amino acid sequence of the encoded protein. As a result, to better understand how the protein is affected by mutation, the sequenced mutant RTase genes were translated *in silico*, and any amino acid changes were scored. The amino acid mutations had less range than the nucleotide mutations, with a maximal number of mutations of six instead of the nucleotide mutation maxima of eight. Similarly to the nucleotide mutation profile, the modal value of mutations was zero, with one mutation then being lower than zero. The proportion of sequences with two amino acid mutations increased from the proportion with one amino acid mutation. The proportion of sequences then decreases as the number of mutations increases. In this way, both profiles resemble a normal distribution skewed towards 0, with additional excess 0 mutations (see figure 4.3.5(A)).



(a)



(b)



(c)

4.3.5 Graphs showing the frequency of mutations found in the RTase mutagenesis experiments. The number of amino acid changes per RTase mutant was calculated. Sequenced mutants were translated *in silico*, and the resultant amino acid sequences were aligned with the wildtype sequence using a Needleman-Wunch algorithm. **(a)** The number of amino acid mutations per amplicon was counted. Additionally, the type and direction of mutation, either with regards to the amino acid identity **(c)**, or amino acid type **(b)** was counted and plotted.

4.3.6 Comparison of PhoEP and Taq/Mn^{2+} mutagenesis

The results from the Oxford Nanopore sequencing of NiFe-PETase data were analysed using a Python program "AnalyseDNAPairwise2.py" (Appendix 2.10). This program ran a pairwise alignment of each sequenced clone against the wildtype sequence, and counted the number and type of mutations that were present. A small number of clones from the Taq/Mn^{2+} contained large deletions when compared to the wild-type gene. These clones – ten in total – were removed from the analysis as these deletions likely came about due to large template slippage, and as to not skew the data with large numbers of indel mutations. These sequences are discussed at length later in the chapter.

The data show that the mutagenesis with PhoEP yield a significantly fewer mean number mutations than mutagenesis with Taq/Mn^{2+}, with a maximal number of mutations of five from the PhoEP mutagenesis, compared to twenty-one from the Taq/Mn^{2+} mutagenesis (see figure 4.3.6(A)). These data also show that the Taq/Mn^{2+} sequences have more variance, with a standard deviation of 4.29 mutations per clones compared to the PhoEP sequences standard deviation of 1.15 mutations per clone.

The type of mutational profile introduced by PhoEP EP-PCR and Taq/Mn^{2+} EP-PCR was also shown to vary. In PhoEP the most common type of mutation to occur was a G to A, or C to T mutation. Owing to the double stranded nature of DNA, it is not possible to know whether the mutation occurred when an adenine was incorrectly incorporated in place of a guanine, or if a thymine was incorporated in place of a cytosine with these data. This is in contrast to the most common mutation in the Taq/Mn^{2+} system, which was a A to G, or T to C mutation (see figure 4.3.6(B)). The least common mutation in both systems was an insertion mutation, with an C or G insertion only representing 0.004 of the total number of mutation introduced in PhoEP, and a A or T insertion representing only 0.0004 of the total number of mutation in the Taq/Mn^{2+} system.

While the number and identity of the mutations varied, the type of mutations introduced followed similar proportions for mutations introduced by PhoEP and Taq/Mn^{2+}; for both systems transition mutations were the most probable type of mutation to occur, representing a proportion 0.63 or 0.70 of the total mutations introduced by PhoEP or Taq/$Mn^{2=}$ respectively. Transversion mutations were less common in both systems,

representing 0.23 or 0.25 of the total mutations in PhoEP or Taq/Mn^{2+} respectively, with indel mutations being least common, at 0.14 or 0.05 of the total mutations (see figure 4.3.6(C)).

Additionally, the sequencing results of the Taq/Mn^{2+} showed that there were 10 mutants with extensive deletions. Nine of these deletions were in the same location – from wild-type ORF position 109 to 759- while the last one with a deletion from position 957 to 1346.

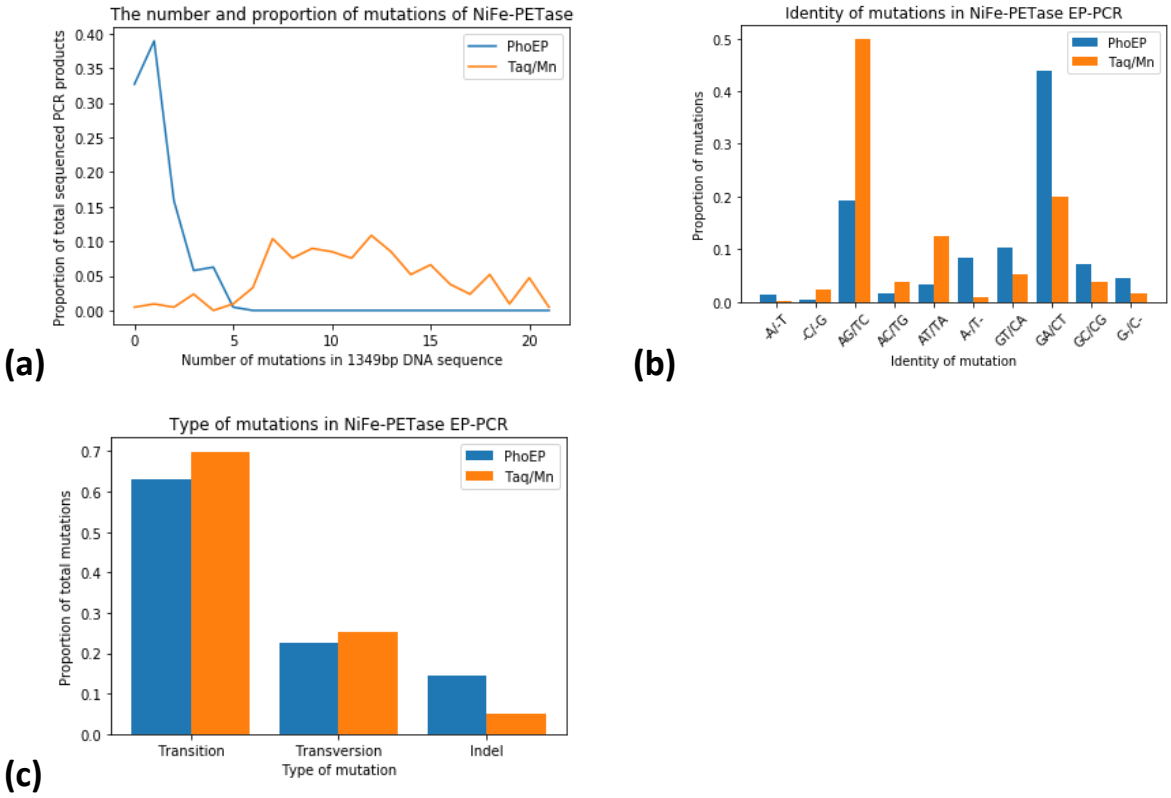


Figure 4.3.6: An analysis of the number and type of mutations introduced over the course of an EP-PCR by PhoEP, or by Taq/Mn^{2+}. (a) The proportion of each number of mutations per 1349bp PCR product was calculated and plotted. **(b)** The proportion of each identity of mutation was calculated. As it is not known whether each mutation occurred on the coding or non-coding strand of DNA, both possibilities are collated together. **(c)** The proportion of each type of mutation is characterised and plotted on a bar chart.

4.3.7 Non-linear least squares analysis of PETase sequencing data

Similarly to the sequences obtained for NiFe and RTase, non-linear least squares analysis was also performed to elucidate the mutation rate of the PhoEP, and the Taq/Mn^{2+} EP-PCR systems. A Python program was written to carry out the non-linear least squares analysis (see appendix 2.9).

Initially the least squares analysis was run assuming perfect amplification efficiency. This yielded mutation rates of 4.59 x 10^{-5} mutations per base for the PhoEP system, and 4.86 x 10^{-4-} for the Taq/Mn^{2+} system (see figures 4.3.7(A) and (D). This rate of mutation is unaffected for PhoEP sequences when we consider imperfect amplification efficiency as a separate parameter to be optimised, as the non-linear least squares analysis found that the model fit the data best when the amplification efficiency was perfect at 0.5. This is opposed to the Taq/Mn^{2+} data, that decreases amplification efficiency to 0.43, with the mutation rate increasing to 7.03 x 10^{-4} (see figures 4.3.7(B) and (E).

Using data collected during the experiments, the average amplification efficiency of the EP-PCRs can be estimated. Namely, the controlled input of plasmid DNA and measurement of purified PCR product can be used to estimate the amplification efficiency. As plasmid DNA was used as the template in both PCRs is larger than the amplicon, the input mass of plasmid first has to be converted to molar input. This input of 1 ng of DNA corresponds to 234 attomoles of 6585 bp DNA assuming a base duplex molecular weight of 649 Da. The purified DNA products were measured on a nanodrop spectrophotometer, giving a mass per µl that could then be converted into a molar quantity using the same formula and with a DNA length of 1593 bp. This resulted in a product molarity of 267 femtomoles and 1.48 picomoles for PhoEP and Taq/Mn^{2+} systems respectively. When these values are used alongside equation 4 from chapter 3.3.3, the resultant amplification efficiency values are 0.1822 and 0.2212 for PhoEP and Taq/Mn^{2+} respectively. When these values are hard coded into the non-linear least squares analysis, the resultant mutation rate of the PhoEP and Taq/Mn^{2+} increases to 1.89 x 10^{-4} mutations per kilobase of DNA extended, and 1.58 x 10^{-3} mutations per base of DNA extended respectively (see figures 4.3.7 (C) and (F).

The cost of the fitted data varied with the changing mutation rate and amplification efficiency. For PhoEP, the optimal cost was achieved with an amplification efficiency of 0.5, and a mutation rate of 4.59 x 10^{-5} mutations per base at 2.20 x 10^{-3}. This cost value

increased when using the calculated amplification efficiency of 0.1822 and mutation rate of 1.89×10^{-4} mutations per kilobase extended to 3.45×10^{-3}.

Similarly, for the Taq/Mn^{2+} system the optimal cost was achieved when the amplification efficiency was varied as a parameter in the non-linear least squares analysis alongside the mutation rate. Assuming a perfect amplification efficiency of 0.5 alongside gave a cost value of 3.41×10^{-3}, whereas when the amplification efficiency was fitted to a value of 0.35 the cost value decreased to 3.15×10^{-3}. The highest cost value for the Taq/Mn^{2+} system was achieved with the calculated amplification efficiency of 0.22, which yielded a cost value of 4.56×10^{-4}.

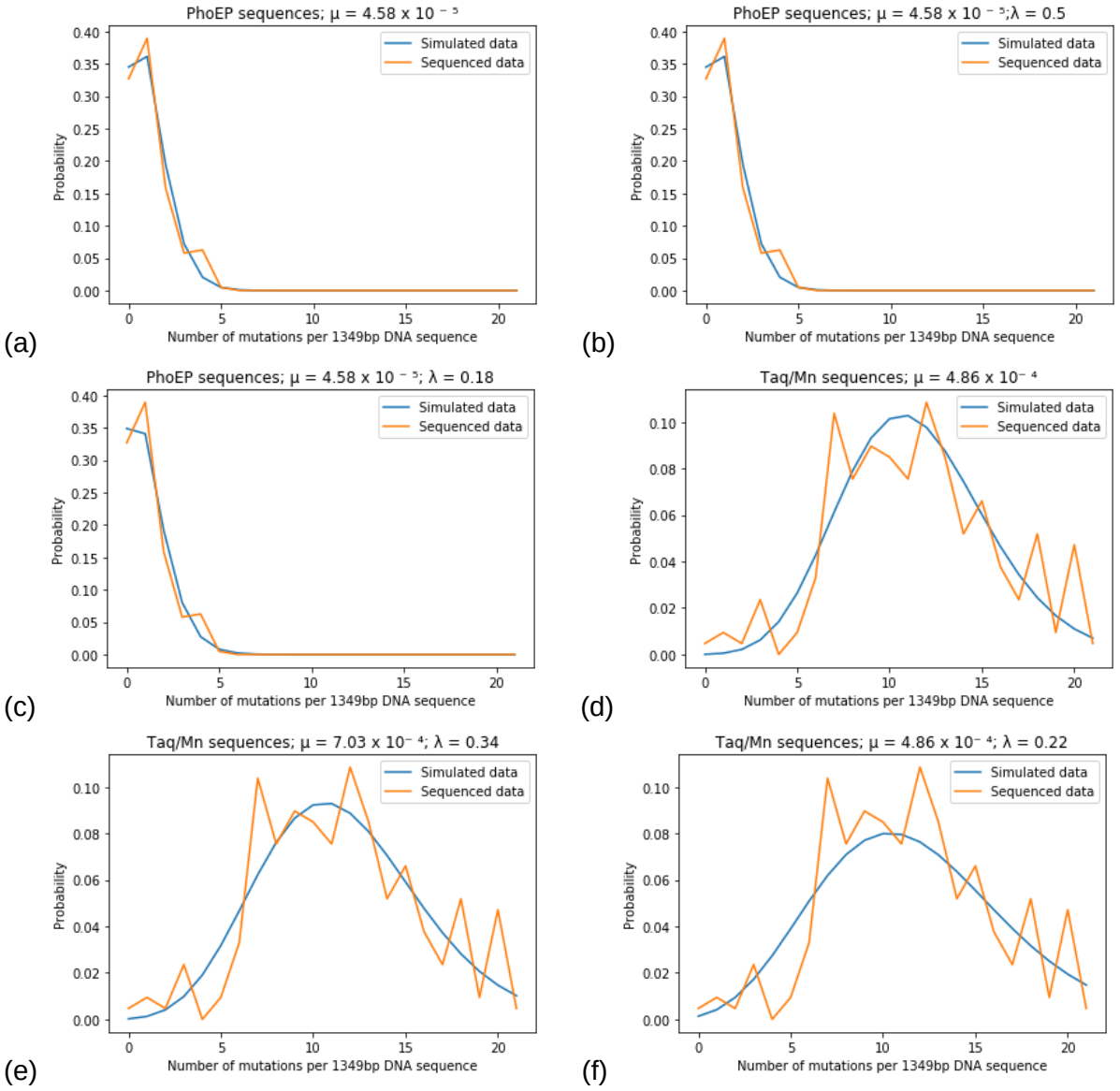


Figure 4.3.7: Non-linear least squares (NLLS) analysis was used to find the optimal mutation rate, or mutation rate and amplification efficiency to fit the modelled data to the sequenced data. These fitted parameters were then used to simulate EP-PCR data (blue lines) (a) NLLS carried out with assumed perfect amplification (λ=0.5) on PhoEP NiFe-PETase data **(b)** NLLS carried out optimising both amplification efficiency (λ) and mutation rate (μ) **(c)** NLLS carried with calculated amplification efficiency (λ=0.18) to fit the mutation rate of PhoEP data. **(d)** NLLS performed to fit the 3.3.4 model to the Taq/Mn^{2+} data assuming perfect amplification efficiency (λ=0.5) **(e)** NLLS performed to fit 3.3.4 model to the Taq/Mn^{2+} data with no assumptions **(f)** NLLS performed to fit the Taq/Mn^{2+} data assuming the calculated amplification efficiency (λ=0.22)

4.3.8 Comparison of EP-PCR models from literature

EP-PCR models previously described in the literature were replicated in Python programs and assessed to compare the accuracy and suitability of the models to the sequenced data. The models described in Wang et al., (2000), Pritchard et al., (2005), and Moore and Maranas (2000), were replicated in a Python program (see appendix 2.9). These programs resulted in probabilities of how many mutations would be introduced for any given rate of mutation and amplification efficiency. As the amplification efficiency in the models was defined differently to in this thesis (being the proportion of DNA strands that make cohesive product per cycle rather than the proportion of DNA strands in a cycle that were made in the previous cycle), the amplification frequency was adapted to match. Additionally, the Moore and Maranas model does not allow for the incorporation of imperfect amplification efficiency, and so this parameter is kept constant in the non-linear least squares analysis of the Moore model. Finally, the Moore model and the Pritchard model utilise a per cycle mutation rate as opposed to the per base pair mutation rate used in the Wang mode and the model outlines in 3.3.4. As such, the mutation rates obtained from these models are divided by the length of the PCR DNA product (1349bp in this instance) in order to normalise the data for length of amplicon.

The data from these models were fitted to the sequencing data using non-linear least squares analysis, in order to ascertain which model was yielded results closest to the sequenced data. As the true value of the mutation rate and amplification efficiency is unknown, the data were analysed by comparing the residuals of the fitted modelled data to the sequenced data.

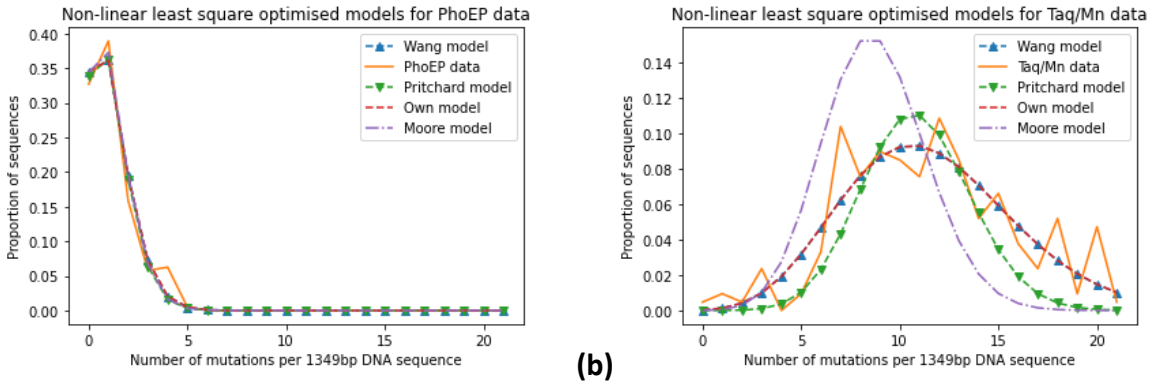
The results of this fitting to the PhoEP data yielded very similar results for all four of the tested models. The model described in section 3.3.4 previously predicted an almost perfect amplification efficiency for these data, which is replicated by the results of the Wang model. As the Moore model does not allow for imperfect amplification efficiency, the mutation frequency of these first three models were very similar at around 4.5×10^{-5} mutations per kilobase extended (see table 4.3.8) owing to their calculation or assumption of perfect amplification. The Pritchard model varied from the other models, having an optimised

average amplification efficiency of 0.147. This lower average amplification efficiency yielded a higher optimised mutation frequency, at 1.72 x 10^{-4} mutations per kilobase extended.

Of the four models tested, the Pritchard model performed best, having an absolute loss cost of only 2.00 x 10^{-3}, compared to 2.09 x 10^{-3-} for the Moore model, and 2.20 x 10^{-3} for the Wang model and the model described in 3.3.4.

The results of the non-linear least squares analysis were different for the Taq/Mn^{2+} dataset; while the Wang model and the model outlined in 3.3.4 resulted in largely the same modelled data once optimised (see figure 4.3.8(B); red and blue lines), the Pritchard and Moore models gave less accurate representations of the sequenced data. This is highlighted by an increase in absolute loss in the Moore model and Pritchard model, at 1.74 x 10^{-2} and 6.21 x 10^{-3} respectively compared to the Wang model and the 3.3.4 model, both with an absolute loss value of 3.14 x 10^{-3}.

These data suggest that at higher rates of mutation, the Wang model and the model outlined in 3.3.4 give a more accurate estimation of the probability that a randomly selected sequence will have *n* mutations from the template sequence compared to the Pritchard model and Moore model. However, at lower rates of mutation the opposite is true, with the Pritchard model performing best, and the Wang model and 3.3.4 model performing worst.



(a) **(b)**

Figure 4.3.8: Models from existing literature were replicated, and subsequently fitted to the sequenced data from PhoEP and Taq/Mn^{2+} EP-PCR experiments.

NLLS was used to fit the models outlined in Wang et al., (2000), Moore and Maranas (2000), Pritchard et al., (2005), and chapter 3.3.4 to either **(a)** sequence data from PhoEP mutagenesis of NiFe-PETase or **(b)** sequence data of Taq/Mn^{2+} mutagenesis of NiFe-PETase.

	PhoEP			Taq/Mn^{2+}		
Model	Mutation rate (per kb extended)	Amplification efficiency	Cost	Mutation rate (per kb extended)	Amplification efficiency	Cost
Wang	4.58E-05	1.00E+00	2.20E-03	7.16E-04	5.24E-01	3.14E-03
Pritchard	1.72E-04	1.47E-01	2.00E-03	6.57E-04	5.48E-01	6.21E-03
Moore	4.45E-05	N/A	2.09E-03	3.71E-04	N/A	1.74E-02
3.3.4	4.58E-05	1.00E+00*	2.20E-03	7.02E-04	6.98E-01*	3.14E-03

Table 4.3.8: The optimised values for mutation rate and (where appropriate) amplification efficiency following non-linear least squares analysis, minimising loss compared to either PhoEP sequenced data, or Taq/Mn^{2+} sequenced data. Amplification efficiencies for the model described in 3.3.4 have been adapted to be in line with other model's definitions.

4.4 Discussion

The results of the sequencing attempts of both RTase and NiFe go some way to validate the statistical model of EP-PCR, however some shortcomings still remain. Because the model predicts a probability distribution, and that EP-PCR generates a large number of sequences, the amount of data required to validate the model is considerable. Using a sample size calculation for an unlimited population, to obtain a 95% confidence interval, at least 385 samples would be required.

Furthermore, as the data-pool is very large, many sequences would have to be analysed in order to carry out a valid sampling. These number of samples would have to be carried out for each different EP-PCR reaction, as each of the distributions would likely be different. Further sequence generation will be needed in order to fully validate the statistical model. It is anticipated that these data will be obtained on an ongoing basis in the supervisor's laboratory.

The work carried out in this PhD supports the prediction of the model that the distribution of number of mutations after an EP-PCR follows a skewed Gaussian model. This is, however, a common distribution and the data might be predicted more accurately by a different model.

The results of the RTase and NiFe mutational analyses showed that the mutant amplicons were most often not mutated by the PhoEP methodology; the modal number of mutations in the 10, 20, and 25 cycle Rubredoxin sequences, and in the 25 cycle RTase sequences was 0. This might be a genuine finding, indicating that in any EP-PCR amplicon quasi-species there will always be a significant population of 0 mutations. Alternatively, this excess 0 value might be due to experimental design.

After a PCR has reached completion, the original template will still be present in the sample. If the amplification efficiency of the PCR is low, or the concentration of template is very high, this could mean that the dominant DNA species in the product is either the template, or a low generation number amplicon. Both of these results would have a relatively low number of mutations in comparison to amplicons with a higher generation number. This hypothesis could be tested in a few different ways: firstly, the EP-PCR product could be treated with the modified DNA restriction enzyme Dpn1. This would degrade any adenine methylated DNA at GATC sequences, meaning that any template plasmid DNA would be

removed from the amplicon library. Doing so would allow identification of whether excess template is influencing the results. Alternatively, a series of EP-PCRs could be set up diluting the input template as much as possible. This would allow the minimum possible quantity of template that could be added while generating a viable downstream product to be ascertained. This minimum template could then be used to generate a mutant library, which could be sequenced, and compared to the results reported in this thesis. This comparison would clarify whether the excess 0 mutations are inherent to the system or are due to excess template/low generation number mutants.

During sequencing of PETase sequences, no template carryover occurred from the PCR, by experimental design. This was achieved by either excising a specific band following agarose gel electrophoresis of the PCR product, or by purifying the PCR product and digesting with DpnI to selectively degrade *dam*-methylated DNA (Geier and Modrich, 1979). These experiments did indeed result in fewer unmutated PCR products; however, this might have been due to the increase in the cycle number in the latter EP-PCR experiments up to 35 cycles. To this end, the optimised parameters from the NiFe and RTase experiments were used with the model from 3.3.4 to simulate data from a 35 cycle EP-PCR (see figure 4.4.1). The results showed that in most cases the excess unmutated PCR products were no longer the modal population. The exception to this was the parameters from 20 cycle NiFe sequences, which had a low very low amplification efficiency – at 0.03 - suggesting that there will be excess unmutated PCR products if the amplification efficiency is low.

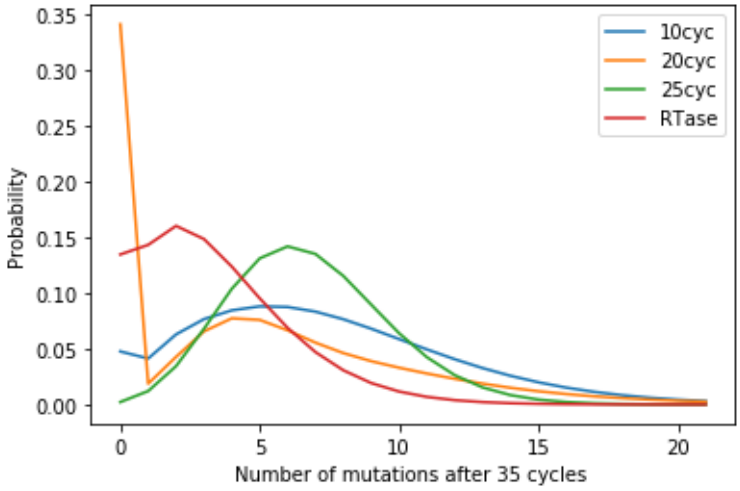


Figure 4.4.1: NLLS parameters from earlier sequencing work were used to extrapolate the mutational probabilities after 35 cycles. The data show that the excess unmutated sequences are not present, with the exception of parameters derived from 20 cycle NiFe NLLS.

The generation and sequencing of large numbers of PETase sequences allowed a greater understanding of the number and type of mutations introduced by both PhoEP and Taq/Mn^{2+} systems. While the Taq/Mn^{2+} system was vastly more mutagenic than the PhoEP system – yielding a ten-fold increase in the mean number of mutations introduced in the sampled population – both systems gave similar results for the distribution of mutation type, with transitions being most common, and indels being least common. These results mimic those found in nature, with transversions being less commonly occurring than transitions (Fitch, 1967).

Ten NiFe-PETases mutated by Taq/Mn^{2+} method were found to have large deletions. Despite nine of these mutants sharing the same large deletion, they additionally contained other mutations differentiating them from one another; among the nine mutants, there were six unique mutational profiles wherein the base substitution mutations were completely distinct. There were two sets of identical mutations, one with three clones and one with two clones. This lends credence to the thought that mutations created in earlier generations of DNA strands are carried forward to their "offspring" DNA strands, providing evidence for the theory of generations originally outlined by Sun (Sun, 1995). This is because it is more likely that this large deletion occurred once, and each of these mutants are an offspring of a theoretical progenitor DNA strand, than it is that exactly the same deletion occurred individually on six occasions.

The fitting and comparison of different models to the sequence data confirmed that there is no “perfect” model for the introduction of mutations via EP-PCR. While the Wang model and the model presented in this thesis were most accurate in the modelling of Taq/Mn^{2+} data, the Pritchard model was most accurate in the modelling of PhoEP data. Interestingly, the Pritchard model was the only model that correctly fitted a sub-optimal amplification efficiency for the PhoEP NiFe-PETase dataset. This reduced amplification efficiency also results in an increased mutation rate compared to the data fitted by the other models. This increased rate of mutation is also closer to the value obtained when analogous mutations were made in *Pfu* polymerase (Biles and Connolly, 2004). These details combined suggests that the Pritchard model is most accurate at predicting the introduction of mutations in an EP-PCR system utilising PhoEP.

The concepts behind the models examined in this thesis – such as generation – were originally conceived in modelling PCR rather than EP-PCR, and as such might be inaccurate when larger mutation rates are considered (Sun, 1995).

It is clear from previous literature and work carried out in this thesis that EP-PCR is an intrinsically stochastic process; the probability distribution of mutations obtained in one EP-PCR will not necessarily be observed in future experiments, even if all parameters are kept the same. Pritchard et al (2005), showed this to be true when repeatedly modelling EP-PCR, and sampling either 10 or 100 randomly selected sequences. They found that the normalised hamming distance between the sampled sequences varied from 0.045 to 0.060. They found that some of this variance was due to the differences GC content of their chosen genes, due to the fact that the mutation matrix that they used favoured conversion of G and C nucleotides (Pritchard et al., 2005).

One reason for these incongruencies between sequenced data and models could be the mutable nature of amplification efficiency and mutation rate within an EP-PCR experiment. There has been some work to incorporate a changeable amplification efficiency into EP-PCR models (Pritchard et al., 2005), along with experimental evidence (Rutledge and Stewart, 2008) that the amplification efficiency of EP-PCR changes over the course of the experiment.

In one of the models outlined in the Pritchard et al., (2005), the amplification efficiency decreases linearly towards 0 as the number of DNA strands increases towards an arbitrary limit. This acts as a good first step in the incorporation of a changeable amplification efficiency but falls short of the complex sigmoidal nature of amplification frequencies often observed in experimental data. Additionally, the concept that the mutation rate changes as the EP-PCR proceeds has also been noted and incorporated into EP-PCR models (Moore and Maranas, 2000), with it being generally noted that mutation rates will change depending on the availability of each free dNTP in the EP-PCR reaction. However, there has been no experimental data suggesting how the mutation rate changes.

A future direction this field could take would be to carry out cycle-by-cycle analysis of both mutation rates and amplification efficiencies. This would allow a much more robust view of how these key parameters change over the course of an EP-PCR. It is likely that a study like this has not already been undertaken due to the large scope of this potential work, and due to the stochastic nature of EP-PCR making like for like comparison difficult. However, by following a protocol similar to that laid out by Karst et al (2021), it may be possible to retrieve a large number of EP-PCR products on a cycle-by-cycle basis. This theoretical protocol would be to take a sample of the EP-PCR mix after each cycle and put it into a secondary PCR containing unique molecular identifier (UMI) tagged primers. This would allow individual PCR products to be sequenced at sufficient read depth to have confidence in the results of the sequencing.

5. Functional assay and screening of random mutant libraries

Abstract

While the number of mutations that are introduced over the course of an EP-PCR is important and essential to maximising the probability of attaining a successful mutant, the number of mutations is ultimately irrelevant in comparison to the ability of the enzyme to carry out its function. In this chapter, wild-type reverse transcriptase was expressed, extracted and purified from *E. coli* in order to provide a benchmark for a large scale directed evolution programme. The mutant library generated in chapter 4.3 was expressed in *E. coli* and the resultant recombinant RTases were extracted, and subsequently tested for activity. A selection of these enzymes was then assayed at elevated temperatures in an attempt to identify any thermostable RTase mutants. It was found that the majority of mutants assayed had reverse transcription activity when assayed at 55℃ by end-point RT-PCR. These results highlight the robustness of the RTase sequence and illustrate how much it can be mutated while retaining functionality.

5.1 Introduction

There have been previous successes in the directed evolution of RTase (Oscorbin and Filipenko, 2021). As an essential component of many research and diagnostic genetic processes, an efficient RTase is a highly desirable reagent. Research into RTases has primarily been aimed at increasing its thermostability in order facilitate the denaturation of RNA secondary structures in templates, thereby making them accessible to RTase and reverse transcription. The most commonly used RTase used in research is derived from Moloney Murine Leukaemia Virus (MMLV), mostly due to the fact that it is monomeric, reasonably robust and is generally easier to purify. However, in recent years, there has been a move towards the use of intron II encoded RTases, which lack a RNAse H domain (Mohr et al., 2013)**,** and are claimed to have higher processivity and lower DdDp activity (Smith et al., 2005) than retroviral RTases.

The rational design of mutations to improve the properties of any protein requires a deep understanding of the biochemistry of that protein's structure and function. Such logical design has been successfully carried out in RTase multiple times. For example, the 24 N-terminal amino acids in MMLV RTase were found to be disordered, and as such might affect

the solubility of the protein. It was found that once these amino acids were removed from the RTase construct, the solubility increased. Das and Georgiadis (2001) then theorised that solubility could be further increased by mutating any solvent exposed hydrophobic amino acids. They found a region of hydrophobic residues in the connection domain of MMLV RTase which, when mutated, substantially increased the solubility of the RTase. This result shows that it is possible to logically design mutations in RTase in order to improve a certain parameter.

In addition to targeted techniques such as site directed mutagenesis (SDM), random mutations have also yielded successes in the improvement of RTase functionality. Arezi and Hogrefe (2009) used random mutagenesis in an attempt to isolate mutants that had a higher thermotolerance than wild-type MMLV RTase. After finding four independent mutants that increased the activity at 52˚C. Arezi and Hogrefe (2009) carried out saturation mutagenesis on these positions in addition to 4 further positions which lay in close proximity, to find the preferred mutation at each of these sequence positions. They then carried out combinatorial mutagenesis to determine whether the influence of these mutations were additive. The outcome of their studies was the discovery of an RTase carrying the mutations E69K/ E302R/ W313F/ L435G/ N454K, that had an increased specific activity at 55°C.

While it is a necessity of evolution that all proteins have some level of mutability, some proteins are more amenable to mutations than others. This can be visualised by the analysis of phylogenetic trees of proteins, wherein distally related species might have genes that are almost completely identical. It is therefore important to consider when randomly mutating a protein how much it can be mutated while retaining some level of functionality.

The idea of error catastrophe was originally posited by Eigen (1971), and cemented in literature in by Eigen and Schuster (1977). This is a theory that suggests that there is a maximal rate at which mutations can occur, and that if this maximal rate is exceeded there will be an overall rapid decrease in the fitness of the organism. This theory has been demonstrated experimentally by Crotty et al. (2001) who showed that the antiviral drug ribavirin acts by increasing the mutagenesis in RNA viruses, and moreover demonstrated this effect in poliovirus. While the concept of an error catastrophe was formulated with respect to a species, or quasi-species, it is valid to expand these ideas to in a more simplistic manner to an individual gene.

5.2 Methods

5.2.1 Expression of recombinant MMLV RTase

A synthetic gene encoding RTase from MMLV was obtained from Eurofins Genomics and was subsequently subcloned into a pET28a vector. This allowed the RTase to be expressed under the control of the T7 promoter, and then to be purified utilising the hexa-His tag encoded by pET28a adjacent to the RT open reading frame. This recombinant plasmid was named pJF012 and was used to transform competent *E. coli* BL21 (DE3) cells.

Following successful transformation, RTase variants were expressed from single colonies scaled up to 25ml broth cultures. Overnight 25ml cultures were routinely grown in 250ml Erlenmeyer flasks in LB supplemented with kanamycin as described in chapter 2. When the culture had reached an OD_{600} value of around 0.6, IPTG to a final concentration of 1mM was added to induce RTase expression. The induction schedule was optimised empirically (see figure 5.2.2), as 4 hours at 25℃. Additionally, substitution of a GroESL strain for BL21(DE3) was investigated but made no significant difference to expression levels.

Various lysis techniques were investigated to optimise cell lysis with retention of RTase activity. These included: using non-ionic detergent solutions such as BugBuster (Merck), NEBExpress Lysis reagent (NEB), NiNTA specific lysis solution (Qiagen) and sonication. It was clear that all methods were equally satisfactory, and in all cases, the yields of soluble, active RTase were much lower than (for example) PhoEP, but sufficient to continue with screening experiments (see figure 5.2.1 (B), lane 6 and 7.

Following induction and lysis, the clarified, lysed extract was either stored at 4˚C, or purified by Ni-NTA chromatography (Qiagen) according to the manufacturer's instructions.

Eluted fractions containing imidazole, were then analysed by SDS-PAGE as shown in figure 5.2.1(C). RTase of the expected molecular weight was eluted in fractions 2, 3, and 4 (see figure 5.2.1(C); lanes 7-9).

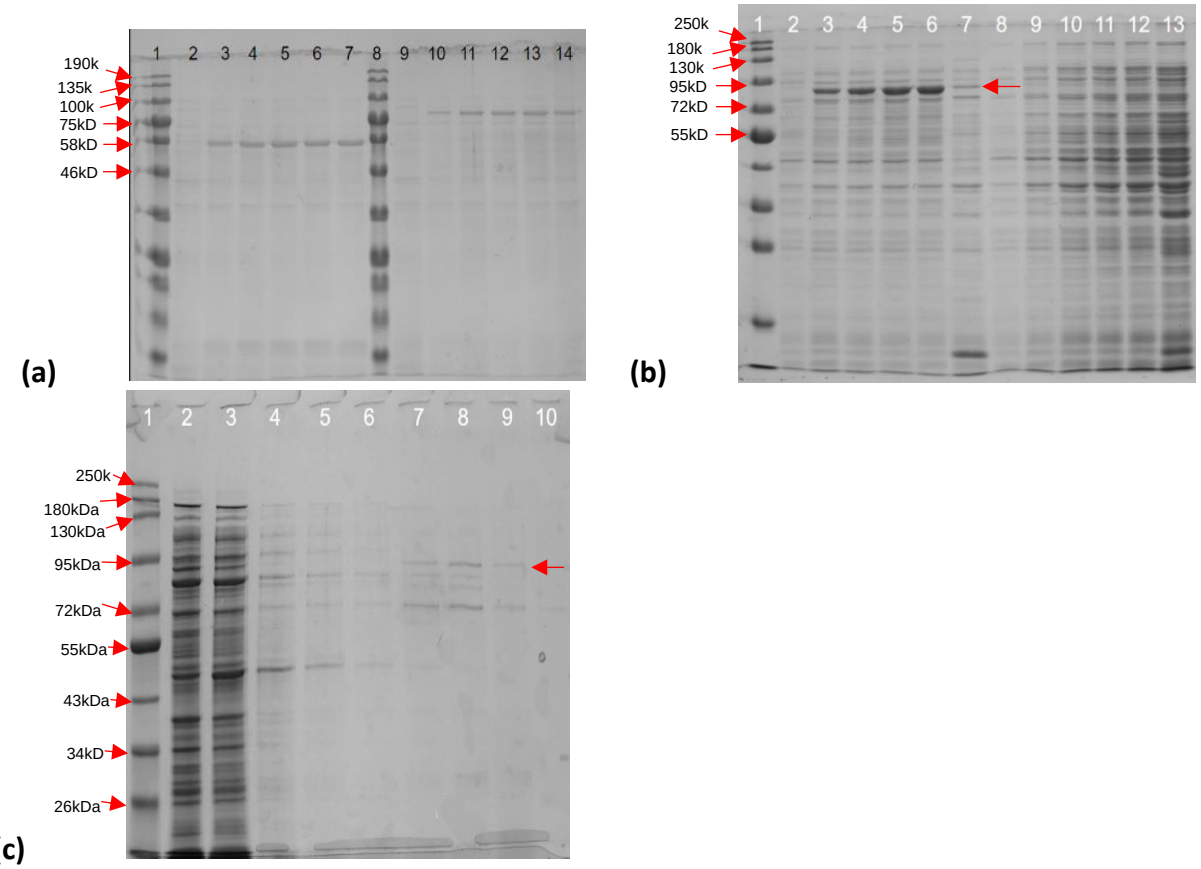


Figure 5.2.1: SDS-PAGE analysis of the expression and purification of recombinant RTase.
(a) GroESL and BL21(DE3) strain *E. coli* containing pJF012 – containing WT RTase – were grown in 25ml LB Kan^{+}. Expression of RTase was induced, and samples were taken from both strains every hour for 4 hours, and at 16 hours. These samples were normalised by OD_{600}, and were subsequently loaded onto an SDS-PAGE. **Lanes 2-7** contain GroESL strain induction of 0, 1, 2, 3, 4, or 16 hours respectively. **Lanes 9-14** contain BL21(DE3) strain induction of RTase at 0, 1, 2, 3, 4, or 16 hours respectively. **(b)** BL21(DE3) strain *E. coli* containing the pJF012 plasmid was grown at 37°C and then either induced with 1mM IPTG, or not induced (**lanes 2-7, and lanes 8-13** respectively). The strains were then grown at 25°C with shaking. 1ml samples were taken from each culture at 0 hours (**lanes 2+8**), 1 hour **(lanes 3+9**), 2 hours (**lanes 4+10**), 3 hours (**lanes 5+11**), or 4 hours **(lanes 6+12**). These timepoints were lysed in SDS lysis buffer. **Lanes 7 and 13** contain the clarified cell lysate following pellet resuspension in NTA binding buffer, and lysis using Lysozyme and sonication. **(c)** After RTase expression and lysis of the cells, the RTase was purified on a Ni-NTA column. 500µl resin volume was used, which was equilibrated with a phosphate-based EQ buffer. 250µl lysate was applied to the column and the flow through was captured as Sample Load fraction (**lane 2**). The column was then washed with 4ml phosphate-based wash buffer and 4x1ml fractions were captured (**lanes 3-6**). 3ml elution buffer was then applied to the column, and the flow-through was captured as 6x0.5ml fraction (**lanes 7-10** represent elution fractions 1-4)

5.2.2 Purification of recombinant RTase

Following confirmation of expression and extraction of RTase, a larger batch (250ml) was prepared for confirmation of the polypeptide integrity using tandem mass spectrometry (MS/MS) analysis. Samples from the large scale preparation were analysed by SDS-PAGE (see figure 5.2.2) and a final purification step of gel filtration and desalting was used prior to MS/MS analysis (see figure 5.2.2(C)).

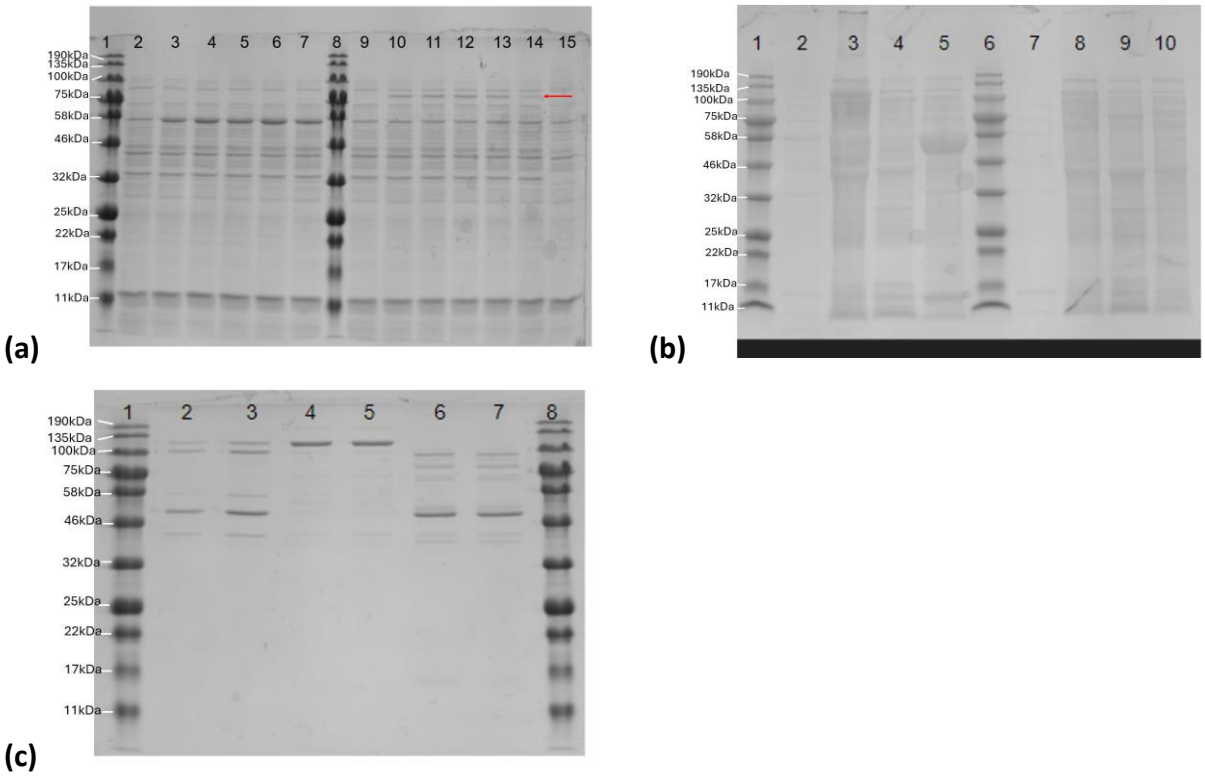


Figure 5.2.2: Expression and downstream purification of RTase (A) SDS-PAGE of induced GroESL pJF012 timepoints (**Lane 2-6**; 0-4 hours, **Lane 7**, 18 hours) and BL21(DE3) pJF012 (**Lanes 9-14,** 0-4 hours; **Lane 15**, 18 hours). **(B)** Elution fractions from Co-IMAC column, wherein lysate from either GroESL pJF012 (**lanes 2-5**) or Bl21(DE3) pJF012 (**lanes 7-10**) were loaded. **Lanes 2-4** were loaded the elution 1-3 fractions resulting from loading of GroESL expressed cell lysate respectively, while **lane 5** was loaded with the cell lysate. **Lanes 7-9** were loaded with elution fractions 1-3 respectively from the BL21(DE3) expressed RTase cell lysate, while **lane 10** was loaded with the cell lysate. **(C)** E2 (GroESL expressed RTase; figure B, lane 3) was then run on a gel filtration column. Fractions corresponding to the largest UV peaks were chosen to be run on an SDS-PAGE.

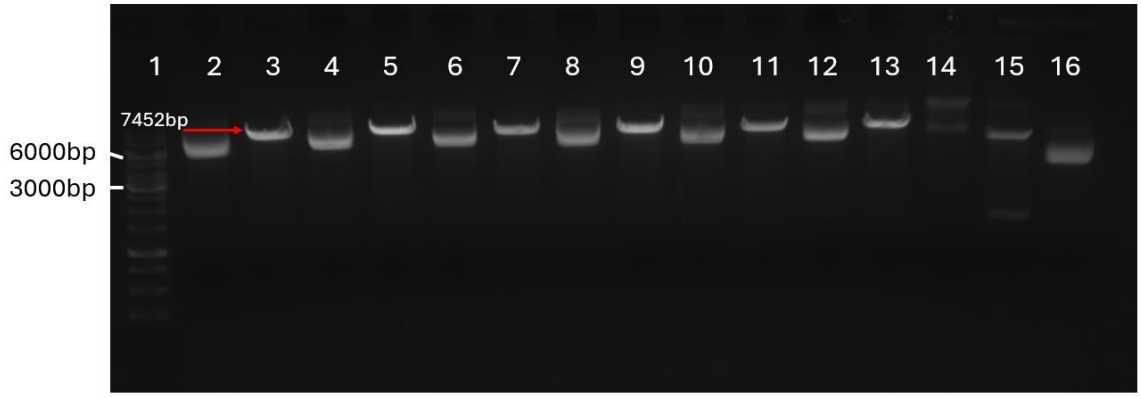
5.2.3 Functional assay of purified RTase

Several different methods were used to confirm the catalytic activity of the purified RTase. Initially, an endpoint RT-PCR was carried out, using yeast RNA templates. Three primer sets were used for this RT-PCR and were designed with help from Dr Ewald Hettema (Department of Molecular Biology and Biotechnology; University of Sheffield)**.** These primers were designed to amplify either the Snc1 gene or the Act1 gene, both of which are spliced in yeast (see appendix 1.1). The primers were designed such that they bridged a splice site, which allowed an RT-PCR to be carried out with the assurance that any amplification products were due to the RTase performing reverse transcription of the spliced RNA rather than the any residual DNA being amplified by the DNA polymerase. Additionally, primers designed to amplify the 18S rRNA, or the full length Snc1 gene present in yeast were designed. These primers would not be able to distinguish between the RNA and any residual DNA left over but were used in order to ascertain whether the RNA extraction and DNase treatment were successful.

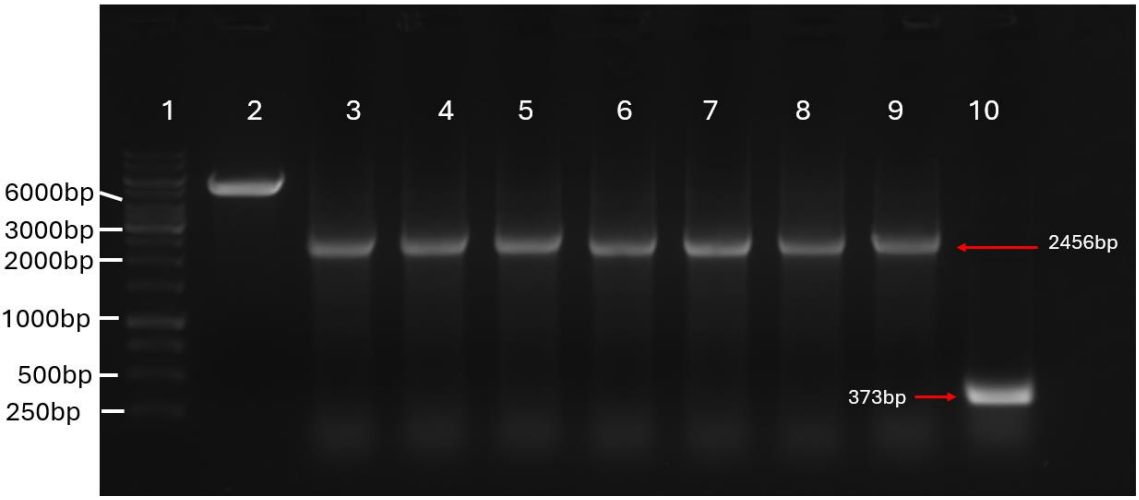
5.2.4 Expression and extraction of mutant RTases from the plasmid library.

Following successful confirmation of WT RTase activity, a mutant RTase library was generated in order to isolate mutants that might have increased thermostability. RTase was mutated following the techniques described in 4.2.4. The mutant amplicon library was subcloned into pET28a, and the resultant mutant plasmid library was used to transform *E. coli* BL21 (DE3) Individual colonies were picked and were grown in LB Kan^+ overnight. Glycerol stocks were then made of these primary cultures for storage of the expression strain. The remaining primary culture was used to extract the plasmid from each strain. These extracted plasmids were then confirmed to contain the RTase gene by endonuclease digestion and PCR (see figure 5.5).

30 clones of the confirmed mutant library were used to extract and assay RTase in culture volumes of 2ml, using the methodology described earlier for the larger scale preparations. Assays were performed by RT-qPCR.



(a)



(b)

FIGURE 5.2.4: Restriction digest and PCR to confirm insertion of RTase into pET28a. (a) A restriction digestion was carried out to confirm that the plasmid extracted from the mutant library did contain the RTase gene. Plasmids were digested using the Eco53kI endonuclease, which would be expected to cut the plasmid a single time, resulting in a band at 7452bp. Digested plasmid was run in **lanes 3, 5, 7, 9, 11, 13, and 15** while undigested plasmid was run in lanes **2, 4, 6, 8, 10, 12, 14, and 16. Lanes 2-13** contained recovered mutant plasmid, while **lane 14 and 15** contained wildtype pJF012 plasmid (pET28a containing RTase gene). **Lane 16** contained undigested pET28a. **(b)** A PCR was also carried using T7F2 and T7R2 primers to confirm the RTase insertion. **Lane 2** was loaded with a restriction digestion of pET28a with Eco53kI – a sample that could not be loaded onto previous gel electrophoresis (A). **Lane 3-8** contains the PCR amplicon product using plasmids extracted from the mutant library as the template. **Lane 9** contains the amplicon using wildtype pJF012 as the template, and **lane 10** contains the amplicon using pET28a as the template. It would be expected that plasmids containing the RTase gene would yield a band at 2456bp, while those lacking the gene would give a band at 373bp.

5.2.5 Catalytic assays of RTase variants derived from the EP-PCR library

The catalytic competence of the mutants in cell lysates was assayed using a ChaiBio Open RT-qPCR instrument. This RTqPCR used synthetic E gene RNA from COVID-19 as the RNA template, and utilised E gene primers in order to facilitate the reverse transcription and subsequent amplification of the gene (Corman et al., 2020). A hydrolysis probe was used to detect the quantity of DNA in the reaction in real time. Each mutant was tested in duplicate.

5.2.6 Thermostability assay of mutant library

Catalytically active RTase variants were assessed for their intrinsic thermostability as follows. Initially, a gradient RT-PCR was carried out in order to find the temperature at which the wild-type RTase loses activity. An end-point RT-PCR was used for this reaction due to the larger capacity of the standard thermocycler, and the ability of the thermocycler to introduce a thermal gradient. The initial reverse transcription stage of the RT-PCR was changed, such that the temperature varied from 55 ℃ to 65. Five mutant RTs were tested alongside the wildtype, and one blank reaction not containing the RTase. The products of this RT-PCR were run on an Agilent Bioanalyzer (see Chapter 2.2.15) in order to validate the molecular weight and concentration of the amplified products.

5.3 Results

5.3.1 MS/MS analysis of RTase fractions

The results of this MS-MS analysis showed several co-purifying polypeptides contained RTase specific sequences. This showed conclusively that multiple bands (namely A and B**)** contained the MMLV RTase protein (see table 5.3.1). These are samples taken from the E2 fraction of the Co-IMAC column, and so indicate that purification of the RTase does occur on IMAC. Additionally, bands G and H were found to have a single peptide that matched MMLV RTase. This could have been due to contamination from a different sample. Additionally, B and G contained protein identified to be GroEL and GroES chaperone proteins. The MS-MS analysis showed that there is definitively RTase expressed in BL21(DE3) strain *E. coli*, and that this RTase is successfully purified using an Ni-NTA column. However, these data also indicate that the RTase is extracted as a heterogeneous mixture of full length and partially degraded polypeptides.

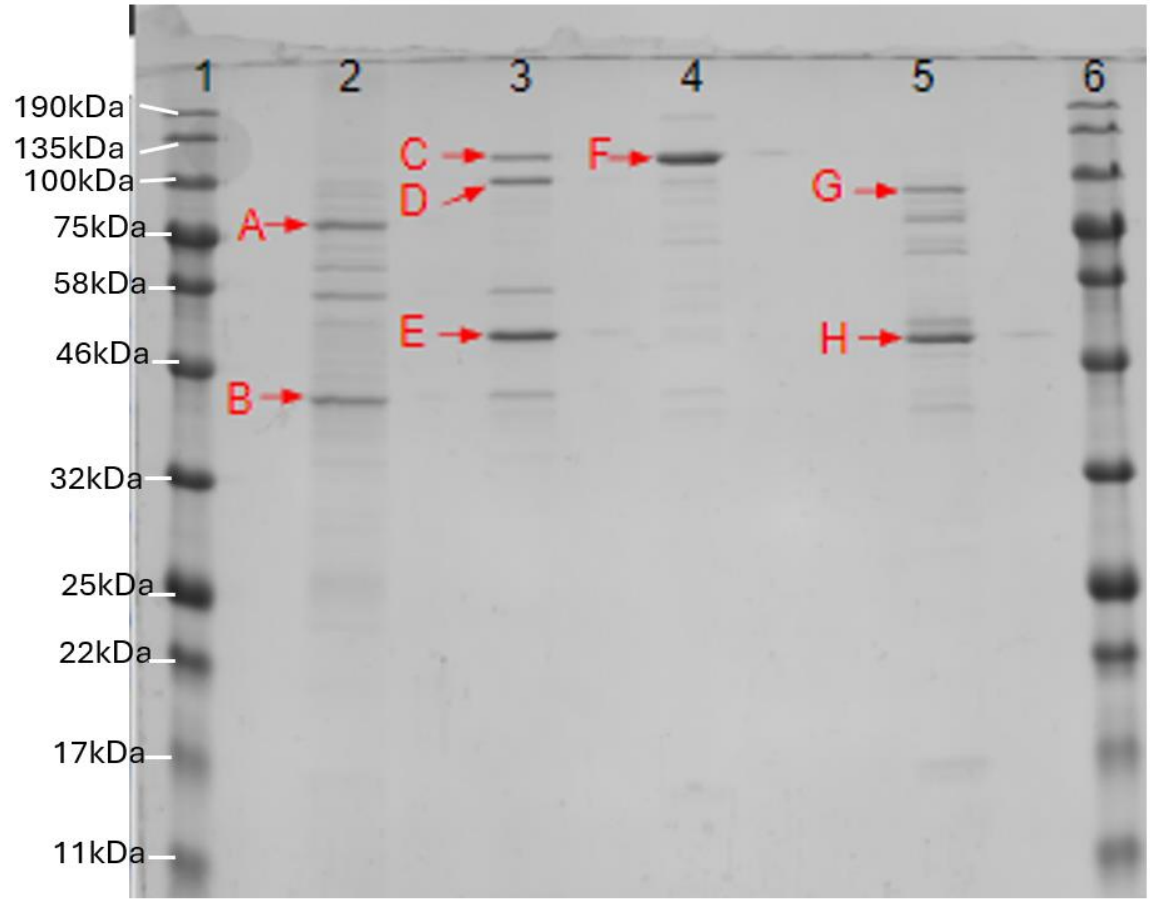


Figure 5.3.1: SDS-PAGE separation of protein samples prepared for MS/MS analysis. Protein samples were loaded onto a 10% SDS-PAGE in order to extract specific bands for a downstream MS-MS analysis. **Lane 2** shows the E2 fraction from a previously described Ni-NTA purification of BL21(DE3) expressing RTase (see 5.2.1). **Lanes 3-5** are specific gel filtration fractions chosen from spectrometric data. Specific bands were then excised and digested with trypsin, before being sent for MS/MS analysis.

		Razor and unique peptides							
Protein Name	Fasta headers	A	B	C	D	E	F	G	H
MMLV RTase	>xx|JF00001| MMLV_RT	53	13	0	0	0	0	1	1
T2 Keratin	>P35908 SWISS-PROT:P35908	38	37	60	46	13	43	47	60
T1 Keratin	>P04264 SWISS-PROT:P04264	37	35	55	39	17	40	45	51
T10 Keratin	>P13645 SWISS-PROT:P13645	35	26	43	36	12	37	40	44
T9 Keratin	>P35527 SWISS-PROT:P35527	29	30	48	40	13	38	42	50
Alanine-tRNA ligase	>sp|P00957| SYA_ECOLI	21	0	0	1	0	0	13	0
Phosphate acetyltransferase	>sp|P0A9M8| PTA_ECOLI	20	0	1	0	0	0	0	0
GTP-binding protein TypA/BipA	>sp|P32132| TYPA_ECOLI	17	0	0	0	0	0	0	0
Ribonucleoside diphosphate reductase 1 subunit alpha	>sp|P00452| RIR1_ECOLI	17	0	0	0	0	0	0	0
T5 keratin	>P13647 SWISS-PROT:P13647	14	11	20	13	4	12	21	21

Table 5.3.1 The most common razor and unique peptide fragments in the MS-MS analysis. From these data it is clear that there is extensive keratin contamination in all samples, but that sample A contains most unique and razor peptides from RTase.

5.3.2 End point RT-PCR for confirmation of RTase activity

An RT-PCR was carried out utilising the extracted yeast RNA as the template and using multiple different RTases in order to carry out the reverse transcription step of the reaction. The samples used were SuperScript II (Thermofisher), clarified cell lysate of expressed RTase (see section 5.2.1), or Ni-NTA purified RTase (see section 5.2.2.). Control reactions contained either all the primers but no RTase, or where a PCR was set up, the "Snc1splice" primers and RNA as a template. Following termination of the RT-PCR reaction, the products were separated and visualised by agarose gel electrophoresis (see figure 5.3.2). The results of these RT-PCRs showed that products of the expected lengths were generated for all the 18S primers, the cell lysate and purified RTase "Snc1DNA" primers, and for the purified RTase "Snc1splice" primers. Additionally, the reaction containing all primers, but lacking RTase has a strong band present at ~800bp, and a fainter band present at ~400bp. These two bands in the no-RTase control reaction are likely due to the 18S and the Snc1 DNA primers amplifying residual DNA in the reaction. The presence of a distinct product when using purified RTase and Snc1 splice primers at approximately 240bp (lane 13) indicated that the purified RTase is successfully catalysisng the RT-PCR reaction. This is further confirmed by the fact that there is no product band present when the Snc1 splice primers were added to a PCR reaction alongside an RNA template in the absence of RTase (lane 15). Due to the absence of any product band in the reactions containing the Act1 primers, the design of these primers was reviewed. It was found that the designed primers were lacking a single base that would be expected at the splice site, which may explain the lack of products in the reactions.

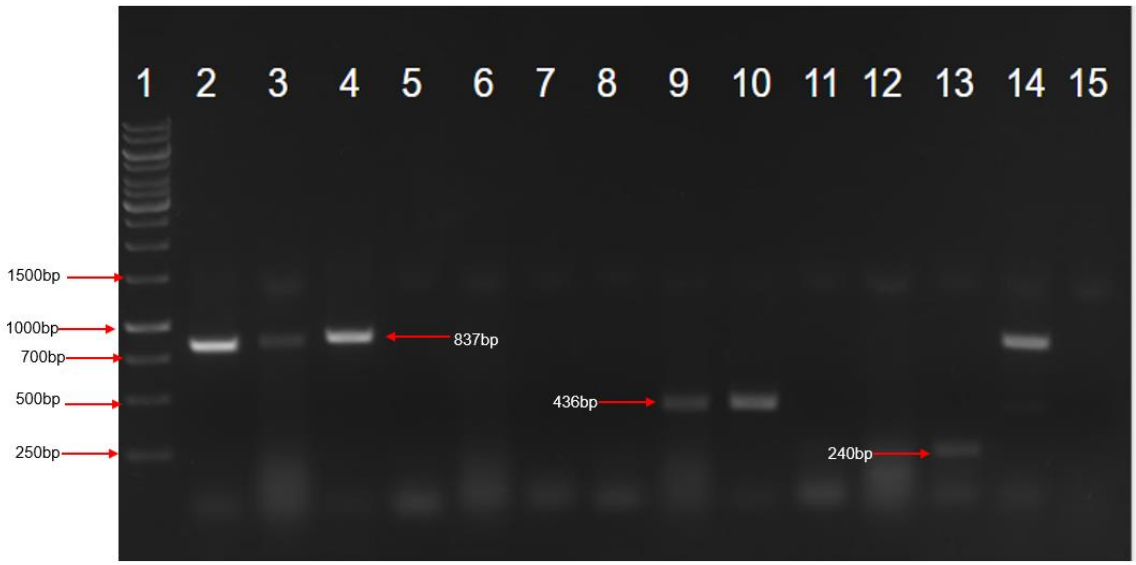


Figure 5.3.2: Agarose gel electrophoresis of the products of end-point RT-PCRS. Four different sets of primers were utilised, targeting four different RNAs; 18S (expected size = 837bp) (**lanes 2-4**), Act1 (expected size = 796bp) (**lanes 5-7**), Snc1 DNA (expected size = 436bp) **(lanes 8-10**), and Snc1 spliced (expected size = 240bp) (**lanes 11-13**). Additionally, further reactions were carried out wherein all primers, but no RTase was added to the reaction **(lane 14**), or where Snc1 splice primers were added to a PCR, utilising an RNA template (**lane 15**). Different RTs were used, with SuperScript II added to the reactions that were run in **lanes 2, 5, 8, and 11**; cell lysate added to reactions in **lanes 3, 6, 9, and 12**; and the purified elution 2 fraction from a NiNTA column added to **lanes 4, 7, 10, and 13.**

5.3.3 Functional assay of RTase mutant library using RT-qPCR

RT-qPCR experiments carried out to follow on from the end-point RT-PCRs showed the majority of the mutant RTases tested were active (see figure 5.3.3(A)). To ensure that this result was not due to non-specific binding of the primers to residual DNA in the cell lysate, the products of the RT-qPCRs were run on an Agilent Bioanalyzer 2100. It was found that all of these products were 120-130 bp in length, which matches the expected length of the amplicon. However, this amplicon was present at low levels in the negative control lacking RTase, suggesting that this band might not be indicative of RTase activity. The band was not present in samples from pJF014A021, or pJF014A029. Both of these samples had lower activity than that of other samples, and of the positive control, suggesting that there might be some correlation between product formation and the relative specific activities of the RTases.

This same assay was performed on the extract of the strain expressing the WT RTase, and the extract from an uninduced strain containing the WT RTase gene. The results of these experiments showed that there was RTase activity in both of these extracts. The proteins present in the cell extracts were then analysed on a Bioanalyzer 2100 protein chip, which showed that there was a band corresponding to the WT RTase even in the uninduced sample. This might be due to incomplete repression of the T7 gene promoter, allowing for low levels of RTase to be expressed, even in the absence of IPTG.

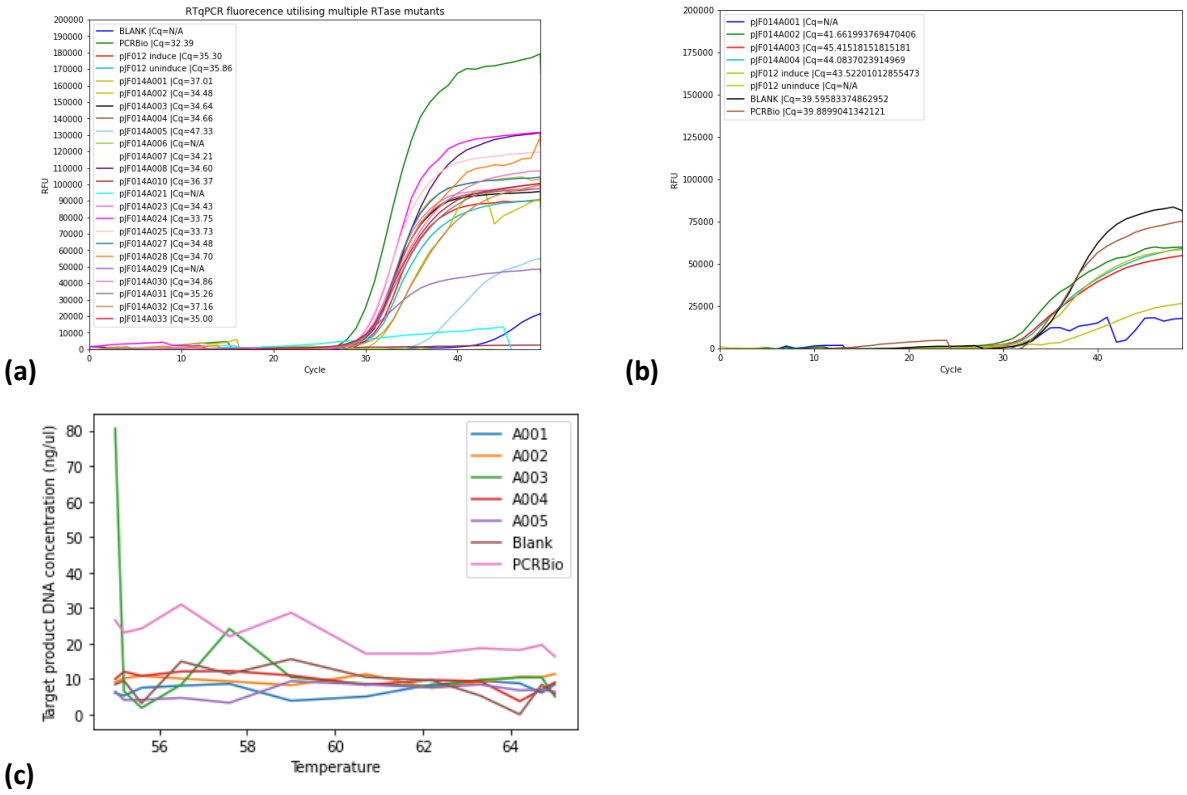
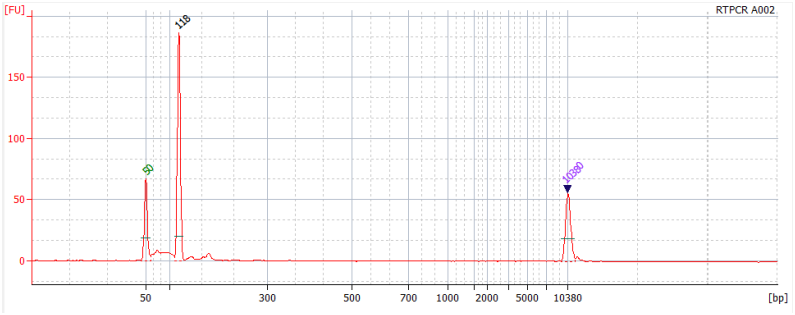


FIGURE 5.3.3 Assessment of mutant RTase activity at 55℃ and 65℃ using RT-qPCR. (a) The RTqPCR results utilising the mutant library of RTase. Mutant RTs were expressed in BL21(DE3) strain bacteria, which were subsequently lysed, and the soluble fraction was extracted. This soluble fraction was used to assay the mutants for activity in an RTqPCR experiment wherein the E gene from COVID-19 was reverse transcribed and subsequently amplified alongside a COVID E gene hydrolysis probe. The fluorescence was measured at each cycle, and the resultant data were plotted on a line graph. The Ct value was calculated by assuming a linear relationship between two cycles and calculating the point at which the RFU would exceed 5,000. **(b)** Multiple mutant RTs were tested to find out if they were thermostable at 65℃. A RTqPCR was carried out as before but holding the mixtures at 65℃ rather than 55℃ for the reverse transcription stage. **(c)** The products from a gradient RT-PCR were run on a Bioanalyzer 2100 DNA 7000 chip, and the concentration of the major DNA species in the sample (120-130bp) was estimated by the software. 5 different mutant RTs were assayed in this way, in addition to a blank sample containing no RTase, and a positive control sample containing UltraScript RTase.

5.3.4 Thermostability assay of RTase mutants

RT-PCR reactions were set up utilising a thermal gradient, increasing the temperature of the reverse transcription from 55 ℃ to 65 ℃ across 12 samples. The RT-PCR products were then run on an Agilent Bioanalyzer 2100 in order to accurately quantify the amount of product in the sample. The results of this gradient RT-PCR showed that all reactions had a similar level of the expected product. This included the blank reaction. As a result, the same reaction was carried as an RT-qPCR, using the highest temperature (65 ℃) as the RT reaction temperature, with the same results (see figure 5.3.3(B). Despite several repeats of these experiments, the same result was obtained.

There are multiple possible reasons that this might have occurred. Excluding buffer contamination with RTase, another possibility is that the Taq in the PCR mastermix is using the intrinsic RdDp ativity present in Taq polymerase (Tse and Forget, 1990) to provide DNA template that can later be used as the template in the PCR reaction. The fact that this activity is observed at 65 ℃ but not 55 ℃ might indicate the inability of Taq to carry out strand-displacement synthesis – a key component of most RdDps like RTase. This would mean that Taq would require higher temperatures to perform the RdDp to eliminate most secondary structures in the RNA. Additionally, the results of this RT-qPCR showed that the final fluorescent values were significantly lower than the previous experiments taking place at 55℃ (figure 5.3.3(A)), and that the Cq values of positive samples were also lower.
The fact that the PCRBio RT positive control sample was consistently higher than all the mutant samples and the Blank negative control might indicate that there was no activity in the mutant samples, and that all activity was due to background Taq reverse transcription activity. This is further suggested by the RTqPCR that was carried out at 65℃ in order to validate these results; the RTqPCR gave much lower relative fluorescence units (RFU) than would be expected – for example comparing figure 5.3.3(A) and 5.3.3(B) it is clear that the fluorescence increase and lag time prior to the exponential phase is almost equivalent to that of RTqPCRs with no RTase activity (figure 5.3.3A, black line). This is further exemplified with the increase of the Cq value from around 32, to around 40 for all active samples.



Band	Size (bp)	Concentration(ng/µl)	Molarity (nmol/l)	Observations
1	50	8.30	251.5	Lower Marker
2	118	16.82	216.0	
3	10,380	4.20	0.6	Upper Marker

Figure 5.3.4: An example Bioanalyzer profile of RT-PCR products and associated peak table. This profile shows the nucleic acid present in the product of an RTqPCR utilising A002 as the RTase. The expected size of the RT-PCR product is approximately 120bp in length.

5.4 Discussion

The successful expression and purification of the WT recombinant RTase was a partial success. The definitive MS/MS data show that the enzyme was expressed as a heterogeneous protein, comprising varying amounts of full length and degraded polypeptides. However, it was felt that in this form, activity assays could be performed with confidence in the presence of suitable controls.

The functional assays of the RTases were also inconclusive. While the preliminary RT-qPCRs showed reasonable evidence for catalytic activity compared to the negative control, the later endpoint RT-PCR, and RT-qPCR at 65℃ showed no difference between the negative controls and samples. Furthermore, both the positive control and negative control gave similar results under after the latter experiments.

The results of the RT-qPCR assays showed that the majority of RTase recombinant extracts were active. Of the 20 clones assayed, only four showed a level of activity in the RT-qPCR assays lower than that of the negative control sample. These samples – A005, A006, A021, and A029 – all had multiple nucleotide mutations resulting in at least one amino acid mutation in each of them.

A005 contained four different amino acid mutations – W172C, Q190R, P354L, and A550S. The Q190R mutation in particular had a marked effect on the ability of this mutant to carry out reverse transcription as it has previously been shown that Q190 (and the analogous Q151 in HIV-1 RTase) are important in the binding of the RTase to the incoming dNTP required for cDNA extension (Halvas et al., 2000). The conversion of this glutamine to a positively charge arginine may prevent dNTP binding, due to the increased positive charge and possible steric clashes with the larger arginine side chain. However, rationalising the negative consequences of the Q190R mutation is difficult in the context of the additional mutations.

A024 only has a single amino acid mutation – R559S. This mutation occurs in the RNase H domain and is not the location of an amino acid that is known critical to nucleic acid binding or RNase H activity. It might be that this amino acid mutation results in unfavourable steric interactions within the RNase H domain, or between domains, that results in the RTase protein misfolding or otherwise adopting a non-functional conformation.

The lack of differences between the different mutant RTase samples indicates a lack of sensitivity and precision with the assay that was used. This might be due to the intrinsic RTase activity of the Taq polymerase used to amplify the cDNA; as it has been reported that under correct conditions, Taq polymerase can extend DNA based on an RNA template (Tse and Forget, 1990), it is possible that the increase in fluorescence is caused by the Taq polymerase amplifying the RNA, and subsequently hydrolysing the hydrolysis probe, causing an increase in fluorescence.

In summary, while the random mutagenesis approach has laid the foundations for a directed evolution study of any protein coding sequence, it does indicate that the greatest challenges lie in the biological expression, solubility, stability and simplicity of the assay used to correlate mutations with protein activity. It is perhaps no coincidence that the search for thermostable variants of MMLV RTases remains a major goal of commercial providers of these enzymes.

6. General Discussion

In the studies presented, a novel method for the statistical modelling of EP-PCR has been developed and implemented. This model has been presented logically and stepwise and has been validated using *in silico* simulations of the EP-PCR reaction, showing that the model allows rapid determination of the probability of the number of mutations introduced, without the need for computationally demanding simulations. However, this statistical model does have limitations, and as such, for more accurate modelling, it is still necessary to perform simulation work.

The statistical model has been tested using the EP-PCR reaction catalysed by a variant of the replication polymerase from *P. horikoshii*. This was followed by the mutation and downstream sequencing of three quite different genes – NiFe, RTase, and PETase. While the results of these sequencing reactions are not definitive, they represent marked progress towards the confirmation of the model and provide some evidence that the model is robust. The sequence data also allowed some characterisation of the PhoEP polymerase, originally developed by Dr Qaiser Sheik (personal communication) and based on the work by Biles and Connolly (2004). The analyses showed that the PhoEP polymerase has a preference for transition mutations over transversions. This is expected, and is in line with natural mutation frequencies (Fitch, 1967). Interesting, it seems that the PhoEP polymerase is also more likely to mutate a cytosine or guanine than an adenosine or a thymine. This result is despite the proportion of CG in both mutated genes being close to 50% (48% in NiFe; 56% in RTase). This might be due to the hydrogen bonding pattern of nucleotides; A and T form 2 hydrogen bonds, whereas C and G form 3 hydrogen bonds. This might mean that a C or a G are less likely to be incorporated in the place of an A or a T due to the fact there would be an unbound hydrogen bond donor or acceptor, resulting in a lower binding strength than in conventional Watson-Crick base pairing.

The comparison of different models of EP-PCR yielded no definitive conclusion on which model represents EP-PCR most accurately. The Pritchard model (Pritchard et al., 2005) fitted the PhoEP mutated NiFe-PETase sequences best, while both the model described in 3.3.4 and the Wang model (Wang et al., 2000) fitted the data from Taq/Mn^2 mutagenesis of NiFe-PETase best. One potential reason for the lack of a clear optimal model could be the fact that none of the models investigated in this thesis utilise variable amplification efficiency or

variable mutation frequency. From qPCR data, it is clear that amplification decreases over the course of the PCR (Rutledge and Stewart, 2008). This has been addressed in some models (Pritchard et al., 2005), albeit in a fairly arbitrary manner utilising a linear decrease in amplification efficiency. A more complex system could utilise data from qPCR in order to accurately follow the rate of decrease in amplification efficiency, possibly by carrying out EP-qPCR experiments. Such a system would likely move away from mathematical modelling, and towards kinetic simulations of EP-PCR based on the previous EP-PCR models and collected sequence and qPCR data.

The additional sequencing carried out in this thesis on NiFe-PETase highlights the mutational differences between different EP-PCR protocols. While the PhoEP system introduced a maximum of 5 mutations and a mean of 1.15 mutations per 1349bp sequence, the Taq/Mn^{2+} system introduced a maximum of 21 mutations, with a mean of 11.43 mutations per 1349bp sequence. These methods of mutagenesis therefore represent two possible rates of mutation that can be introduced over the course of an EP-PCR experiment. This rate of mutation can be changed by varying the cycle number of the EP-PCR, but could also be perturbed by combining the methods. This strategy is already employed by Agilent, who utilise two polymerases in order to fine tune the number of mutations introduced in their GeneMorph kit (Agilent, 2015). By combining these methods at set ratios, and utilising one of the models explored in this thesis, it might be possible to optimise the EP-PCR in order to obtain a desired distribution of mutations in the EP-PCR product.

The random mutagenesis of RTase was ultimately successful, yielding many and various mutants from the wild-type RTase. While the screening of these mutants was fairly inconclusive, it seems likely that only 4 mutants (figure 5.3.3(A); pJF014A006, pJF014A021; pJF014A005; pJF014A029) of the 20 RTases screened had significantly decreased or inactivated RdDp activity. This suggests that MMLV RTase is a fairly robust enzyme, potentially allowing up to 8 mutations while staying active. This is evident from the literature, wherein a plethora of mutations have been found to improve various properties such as processivity, thermostability, and solubility (Oscorbin and Filipenko, 2021).

In Eigen and Schuster's Hypercycle - part A (Eigen and Schuster, 1977), they posit that an error rate of 1% is only "…sufficient to collect and maintain reproducibly an information content not larger than a few hundred symbols…", and further that if the limit error

catastrophe is exceeded that the quasi-species would deteriorate and the entire sequence space would be covered (Tarazona, 1992). These theories are supported by the work carried out in this thesis; despite the fact that the PhoEP (and previously PfuEP) have been mutated and selected to increase the mutagenicity of the polymerases, the estimated error rate in this thesis ranges from 0.0075% to 0.225%. This is a very large range due to the lack of precision in some results, but still remains roughly 5-100-fold lower than that of a theoretical error rate which would cause an error catastrophe when the genome length is a few hundred symbols (Eigen and Schuster, 1977).

Interestingly, in his seminal work, Shannon (1948) utilised information to mean the opposite of Eigen and Schuster in the Hypercycle; while Eigen and Schuster (1977) posit that information content is lost upon error catastrophe, Shannon described the information as the being quantified by entropy, which is the level of "surprisal" in a given message. This meaning that the information content of a message is increased by the lack of knowledge to what is coming next (Shannon, 1948). Therefore, in Eigen and Schuster's example of quasi-species, a system in which no mutations occur has minimal information, while a post error catastrophe system has maximal information as the entire sequence space has been covered.

This concept of Shannon entropy could be used in concert with Eigen's error catastrophe in order to predict the maximal number of mutations that could be introduced before an error catastrophe is triggered. Shannon entropy is already used in some biological systems, such as virology (Wu et al., 2015), wherein high mutation rates result in quasi-species within a single infection. The measure of the Shannon entropy within an infection can be used to predict the recency of infection. Expanding this concept to individual genes, it stands to reason that historical analysis of the entropy any individual gene in a single organism might offer some indication as to the error threshold of that gene in that specific genetic system. If this historical analysis of sequences can be used to estimate the error threshold for genes, this threshold could then be matched by varying parameters in EP-PCR, such that the maximal number of mutations can be attained while retaining function in most of the mutant library.

6.1 Future work

There is much future work that could be done to further develop the theoretical and experimental findings presented in this thesis.

6.1.1 Statistical modelling

While the statistical modelling of EP-PCR did provide some new insights and understanding of how mutations are introduced in an EP-PCR, it does have significant limitations. The most glaring of these is the fact that the amplification efficiency must stay constant over the course of the reaction. This is a large abstraction from reality, wherein it has been often shown that the amplification frequency decreases and plateaus as the reaction proceeds (Liu and Saint, 2002). This decrease in amplification efficiency would likely act to decrease the overall number of mutations in the resultant amplicon library; as the amplification efficiency is decreased towards the end of the reaction, fewer amplicons with high generation numbers would be generated, which is correlated to the length of genetic information extended in the EP-PCR. This would mean that there would be an overall decrease in genetic information extended, meaning that the number of mutations would decrease.

Linked to this is the high probability that the mutation rate does not stay constant over the course of an EP-PCR. Due to thermal cycling of the polymerase and depletion of the dNTPs, it is highly likely that the mutation rate will vary over the course of the reaction. These changes would likely cause an increase in mutation rate, thereby increasing the number of mutations introduced towards the end of the EP-PCR. This concept has been previously noted, in EP-PCR models where the mutation data matrix changes over the course of an EP-PCR (Moore and Maranas, 2000).

The scale of the effects of variable rate of mutagenesis and amplification efficiency is not known, and it might be possible that these two limitations might act to negate each other to a certain extent. The introduction of these two variable parameters would probably increase the accuracy of the model, however more work would have to be done to replace the binomial distribution (chapter 3.3.3) with a different distribution that allows varying event probabilities. Additionally, while there is a lot of literature supporting the decrease in amplification efficiency (Rutledge and Stewart, 2008), there is less to support the increase in mutation rate, bar the evidence that decreasing dNTP concentration increases random

mutagenesis (Fromant et al., 1995). Therefore, more experimental data would need to be collected pertaining to the scale and direction of the change in mutation rate in EP-PCR experiments. This could be achieved by carrying out many repeats of different cycles of EP-PCR, attempting to minimise any decrease in amplification efficiency and optimise amplification efficiency. By sequencing many of each of the different cycles of EP-PCR, it might then be possible to ascertain whether there is any significant difference above what is expected for the different number of cycles. However, due to the stochastic nature or EP-PCR, it might be that an unfeasible number of replicates and sequences would have to be carried out to determine this.

Another issue with the model is that it does not allow for any mutational biases based on sequence. As is evident in chapter 4.3.1 and 4.3.3, the PhoEP polymerase has distinct biases for which base pairs it mutates, and for what base pairs it introduced. This could have a marked impact on the number and type of mutations that are introduced over the course of an EP-PCR. It would therefore be beneficial to implement some of these biases in the model. This would have to be calculated for each polymerase used in EP-PCR and would vary depending on the CG content of the amplicon of interest, however, could improve the accuracy of the model significantly.

6.1.2 Validation of the statistical model

In addition to the limitations in the theory behind the model, more work could also be done in validating that the model is accurate, and correctly predicts the number of mutations in any EP-PCR. To this end, primarily more sequencing would need to be carried out. This could be an extension of the work carried out in this thesis, wherein genes are mutated, subcloned into a plasmid vector and subsequently transformed into *E. coli*. This method allows for the recovery of the mutants for downstream validation or functional assay but is not necessary if the sole aim behind the sequencing reaction is the validation of the EP-PCR model. Therefore, it could be beneficial to redesign the mutagenesis and sequencing experiment, such that a higher throughput methodology could be implemented.

Using next generation sequencing, a method can be envisaged wherein the total EP-PCR product library is tagged and sequenced using unique molecular identifiers (UMIs). This would follow a similar protocol to one already outlined in the literature, wherein a high-

fidelity polymerase is used for two cycles of PCR in order to integrate UMIs onto the termini of DNA of interest. This PCR product is then purified and used as template in a second high-fidelity PCR, amplifying both UMIs and the genes of interest (Karst et al., 2021). This technique could be applied to EP-PCR products, if the input and output DNA in all PCRs are tightly controlled, allowing for troubleshooting of any issues at any stage of the experiment. This would include ensuring that no template DNA is carried over from the EP-PCR, ensuring that the UMIs are sufficiently long such that there are no duplicate UMIs, and that the input DNA is such that there is sufficient read depth following the Nanopore sequencing.
Next generation sequencing provides a lot of promise in the area of EP-PCR modelling and directed evolution; for example, Oxford Nanopore has released new sequencing chemistry Flow cells (R10.4) and kits in order to improve the read identity accuracy of their product. (https://nanoporetech.com/platform/accuracy, 2022). This should allow for better confidence in sequence data from more shallow read depths, thus allowing more clones to be sequenced at a time. Additionally, there have been multiple improvements made to the basecalling software used in this thesis – Guppy – as well as other software that might be used to further improve the read identity accuracy. (Lee et al., 2021)

6.1.3 Directed evolution of reverse transcriptase

One of the main disappointments of this work was the inconclusive RTase mutant screens. While the initial screen of 25 mutants seemed relatively definitive – all mutants with the exception of 3 were fully active – further analysis and experiments proved that this was not certain as initially thought. Screens varying the temperature of the RT step proved inconclusive, with the mutants and positive control having a similar resultant concentration as the no RTase blank.
The RT-PCR experiments utilising the wild-type RTase showed that there was reverse transcription activity in the clarified cell lysate of the induced expression strain, but that the activity increased when a purified fraction of RTase from a Ni-NTA column was used. This could indicate that more definitive mutant results could be achieved by purifying each of the mutant RTases. This is unfeasible even with the relatively low number of mutants given the current protocol of expression in 25 ml, lysis, and NiNTA on a 1ml column. As such, in order to test this hypothesis, it would be required to change the protocol. One potential

way of doing this would be to grow and express the mutant library in a 1ml 96 well plate. This would allow expression of up to 96 mutants in a single plate. After expression, this plate could be centrifuged, and the resultant pellet could be lysed in the same 96 well plate. Such a protocol has been used before in the literature (Leferink et al., 2019), however this would require optimisation and assay development for RTase activity.

Data from this thesis shows that the RTase is most active once purified on an Ni-NTA affinity column. Following lysate clarification, the mutant RTases could then be purified from the soluble fraction using a IMAC pull down experiment, wherein the IMAC beads could be added directly to the samples, incubated for a sufficient amount of time, and then gently centrifuged, with the supernatant then being discarded. Washes and elutions of the IMAC could be carried out in this manner, resulting in purified RTase mutants. The use of 96 well plates in this potential protocol opens it up to the possibility of automation by liquid handling robot, or just 8-chamber pipettes.

One potential issue with this protocol would be the level of expression; from previous experiments in chapter 5.2 show that the majority of the RTase is currently expressed in the insoluble fraction. This is not a major issue when using 25ml cultures, as sufficient RTase is retrieved from the reaction, however when using <1ml it is possible that insufficient amounts of RTase would be retrieved.

The major missing aspect from the work carried out here is the lack of multiple rounds of mutations, with selection of superior mutants at each round. While the length of the work did not allow for this to be completed, future work could pick up the RTase mutants generated in this course of work, confirm the activity, and carry out another round of mutagenesis. The mutation rate used in this future round of evolution could also be maximised, in order to test the concept of error catastrophe in a single protein. If the error rate is high enough, it might be that a higher proportion of mutants would be inactive.

Perhaps a better future approach for the validation of EP-PCR models is to choose a more favourable target gene-protein. Rapid screening and next generation sequencing (NGS) would be much easier to accomplish using a fluorescent protein, such as a green fluorescent protein. This work is now underway as a follow up to this thesis and will hopefully address the limitations discussed above.

6.3 Closing remarks

While some of the ultimate aims of this project have not been met, the course of work has expanded into unexpected areas. Due to various setbacks, and the climate in which the work was undertaken, less progress than was initially expected was made into the directed evolution and functional assay of RTase. Furthermore, while multiple functional RTases were expressed and sequenced, the course of work ended just at a fairly critical juncture, meaning that the main aim of this work went incomplete.

Additional unexpected progress was made in computational work in the place of “wet” lab work. While the computational aspect of this work was always present, it was expanded upon with beneficial results, such as the development of a statistical error prone polymerase chain reaction.

References

Agilent (2015). GeneMorph II random mutagenesis kit instruction manual. p. 17.

Agilent (2022). DNA 7500 and DNA 12000 Kit for 2100 Bioanalyzer Systems Kit Guide. Https://Www.Agilent.Com/Cs/Library/Usermanuals/Public/G2938-90024_DNA7500-12000_KG.Pdf.

Alexander, G.M., Erwin, K.L., Byers, N., Deitch, J.S., Augelli, B.J., Blankenhorn, E.P., and Heiman-Patterson, T.D. (2004). Effect of transgene copy number on survival in the G93A SOD1 transgenic mouse model of ALS. Mol. Brain Res. *130*, 7–15. https://doi.org/10.1016/j.molbrainres.2004.07.002.

Arezi, B., and Hogrefe, H. (2009). Novel mutations in Moloney Murine Leukemia Virus reverse transcriptase increase thermostability through tighter binding to template-primer. Nucleic Acids Res. *37*, 473–481. https://doi.org/10.1093/nar/gkn952.

Avery, O.T., Macleod, C.M., and McCarty, M. (1944). Studies on the chemical nature of the substance inducing transformation of pneumococcal types: Induction of transformation by a desoxyribonucleic acid fraction isolated from pneumococcus type iii. J. Exp. Med. *79*, 137–158. https://doi.org/10.1084/jem.79.2.137.

Bakhtina, M., Roettger, M.P., Kumar, S., and Tsai, M.D. (2007). A unified kinetic mechanism applicable to multiple DNA polymerases. Biochemistry *46*, 5463–5472. https://doi.org/10.1021/bi700084w.

Baltimore, D. (1970). RNA-dependent DNA Polymerase in Virions of RNA Tumour Viruses. Nature *226*, 1209–1211. https://doi.org/10.1038/2261209a0.

Baltimore, D., Gilboa, E., Mitra, S.W., and Goff, S. (1979). A detailed model of reverse transcription and tests of crucial aspects. Cell *18*, 93–100. https://doi.org/10.1016/0092-8674(79)90357-X.

Barrioluengo, V., Alvarez, M., Barbieri, D., and Menéndez-Arias, L. (2011). Thermostable HIV-1 group O reverse transcriptase variants with the same fidelity as murine leukaemia virus reverse transcriptase. Biochem. J. *436*, 599–607. https://doi.org/10.1042/BJ20101852.

Bateson, W., and Mendel, G. (2011). Experiments in plant-hybridisation / By Gregor Mendel. Exp. Plant-Hybridisation / By Greg. Mendel. https://doi.org/10.5962/bhl.title.4532.

Bell, E.L., Smithson, R., Kilbride, S., Foster, J., Hardy, F.J., Ramachandran, S., Tedstone, A.A., Haigh, S.J., Garforth, A.A., Day, P.J.R., et al. (2022). Directed evolution of an efficient and

thermostable PET depolymerase. Nat. Catal. *5*, 673–681. https://doi.org/10.1038/s41929-022-00821-3.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E. (2000). The Protein Data Bank. Nucleic Acids Res. *28*, 235–242. https://doi.org/10.1038/s41577-020-00473-z.

Biles, B.D., and Connolly, B.A. (2004). Low-fidelity Pyrococcus furiosus DNA polymerase mutants useful in error-prone PCR. Nucleic Acids Res. *32*, e176–e176. https://doi.org/10.1093/nar/gnh174.

Boggy, G.J., and Woolf, P.J. (2010). A mechanistic model of PCR for accurate quantification of quantitative PCR data. PLoS One *5*. https://doi.org/10.1371/journal.pone.0012355.

Bustin, S.A. (2000). Absolute quantification of mrna using real-time reverse transcription polymerase chain reaction assays. J. Mol. Endocrinol. *25*, 169–193. https://doi.org/10.1677/jme.0.0250169.

Bustin, S.A., Benes, V., Garson, J.A., Hellemans, J., Huggett, J., Kubista, M., Mueller, R., Nolan, T., Pfaffl, M.W., Shipley, G.L., et al. (2009). The MIQE guidelines: Minimum information for publication of quantitative real-time PCR experiments. Clin. Chem. *55*, 611–622. https://doi.org/10.1373/clinchem.2008.112797.

Champoux, J.J., and Schultz, S.J. (2009). Ribonuclease H: Properties, substrate specificity and roles in retroviral reverse transcription. FEBS J. *276*, 1506–1516. https://doi.org/10.1111/j.1742-4658.2009.06909.x.

Chelly, J., Kaplan, J.C., Maire, P., Gautron, S., and Kahn, A. (1988). Transcription of the dystrophin gene in human muscle and non-muscle tissues. Nature *333*, 858–860. https://doi.org/10.1038/333858a0.

Chen, K., and Arnold, F.H. (1993). Tuning the activity of an enzyme for unusual environments: Sequential random mutagenesis of subtilisin E for catalysis in dimethylformamide. Proc. Natl. Acad. Sci. U. S. A. *90*, 5618–5622. https://doi.org/10.1073/pnas.90.12.5618.

Clark, J.M. (1988a). Novel non-templated nucleotide addition reactions catalyzed by procaryotic and eucaryotic DNA polymerases. Nucleic Acids Res. *16*, 9677–9686. https://doi.org/10.1093/nar/16.20.9677.

Corman, V., Bleicker, T., Brünink, S., Drosten, C., Landt, O., Koopmans, M., and Zambon Public Health England, M. (2020). Diagnostic detection of 2019-nCoV by real-time RT-RCR.

Carité Berlin 13. .

Crick, F.H.C. (1970). Central Dogma of Molecular Biology. Nature *227*, 561–563. .

Crotty, S., Cameron, C.E., and Andino, R. (2001). RNA virus error catastrophe: Direct molecular test by using ribavirin. Proc. Natl. Acad. Sci. U. S. A. *98*, 6895–6900. https://doi.org/10.1073/pnas.111085598.

Crowther, R.L., Remeta, D.P., Minetti, C.A.S.A., Das, D., Montano, S.P., and Georgiadis, M.M. (2004). Structural and energetic characterization of nucleic acid-binding to the fingers domain of Moloney murine leukemia virus reverse transcriptase. Proteins Struct. Funct. Genet. *57*, 15–26. https://doi.org/10.1002/prot.20224.

Das, D., and Georgiadis, M.M. (2001). A directed approach to improving the solubility of Moloney murine leukemia virus reverse transcriptase. Protein Sci. *10*, 1936–1941. https://doi.org/10.1110/ps.16301.

Das, D., and Georgiadis, M.M. (2004). The crystal structure of the monomeric reverse transcriptase from moloney murine leukemia virus. Structure *12*, 819–829. https://doi.org/10.1016/j.str.2004.02.032.

Diagenode LLC (2019) Cats small RNA-seq kit X24, Diagenode. Available at: https://www.diagenode.com/en/p/CATS-Small-RNA-seq-Kit (Accessed: 23 August 2024).

Drake, J.W., Charlesworth, B., Charlesworth, D., and Crow, J.F. (1998). Estimation of spontaneous mutation rates. Genetics *148*, 1667–1686. https://doi.org/10.1111/1541-0420.00065.

Eigen, M. (1971). Selforganization of matter and the evolution of biological macromolecules. Naturwissenschaften *58*, 465–523. https://doi.org/10.1007/BF00623322.

Eigen, M., and Schuster, P. (1977). The Hypercycle: A Principle of Natural Self-Organisation. Part A: Emergence of the Hypercycle. Naturwissenschaften *64*, 541–565. .

Fitch, W.M. (1967). Evidence suggesting a non-random character to nucleotide replacements in naturally occurring mutations. J. Mol. Biol. *26*, 499–507. https://doi.org/10.1016/0022-2836(67)90317-8.

Fromant, M., Blanquet, S., and Plateau, P. (1995). Direct random mutagenesis of Gene-Sized DNA Fragments Using Polymerase Chain Reaction. Anal. Biochem. *224*, 347–353. https://doi.org/10.1006/abio.1995.1050.

Furman, P.A., Fyfe, J.A., St Clair, M.H., Weinhold, K., Rideout, J.L., Freeman, G.A., Lehrman, S.N., Bolognesi, D.P., Broder, S., and Mitsuya, H. (1986). Phosphorylation of 3’-azido-3’-

deoxythymidine and selective interaction of the 5'-triphosphate with human immunodeficiency virus reverse transcriptase. Proc. Natl. Acad. Sci. U. S. A. *83*, 8333–8337. https://doi.org/10.1073/pnas.83.21.8333.

Geier, G.E., and Modrich, P. (1979). Recognition sequence of the dam methylase of Escherichia coli K12 and mode of cleavage of Dpn I endonuclease. J. Biol. Chem. *254*, 1408–1413. https://doi.org/10.1016/s0021-9258(17)34217-5.

Georgiadis, M.M., Jessen, S.M., Ogata, C.M., Telesnitsky, A., Goff, S.P., and Hendrickson, W.A. (1995). Mechanistic implications from the structure of a catalytic fragment of Moloney murine leukemia virus reverse transcriptase. Structure *3*, 879–892. https://doi.org/10.1016/S0969-2126(01)00223-4.

Gibson, U.E.M., Heid, C.A., and Williams, P.M. (1996). A novel method for real time quantitative RT-PCR. Genome Res. *6*, 995–1001. https://doi.org/10.1101/gr.6.10.995.

Gillespie, J.H. (1981). Mutation Modification in a Random Environment. Evolution (N. Y). *35*, 468. https://doi.org/10.2307/2408195.

González, J.M., Masuchi, Y., Robb, F.T., Ammerman, J.W., Maeder, D.L., Yanagibayashi, M., Tamaoka, J., Kato, C., González, J.M., Masuchi, · Y, et al. (1998). Pyrococcus horikoshii sp. nov., a hyperthermophilic archaeon isolated from a hydrothermal vent at the Okinawa Trough.

Grandgenett, D.P., Gerard, G.F., and Green, M. (1973). A single subunit from avian myeloblastosis virus with both RNA-directed DNA polymerase and ribonuclease H activity. Proc. Natl. Acad. Sci. U. S. A. *70*, 230–234. https://doi.org/10.1073/pnas.70.1.230.

Griffith, F. (1928). The significance of penumococcal types. J. Hyg. (Lond). *27*, 8–159. https://doi.org/10.1017/S0022172400040420.

Halvas, E.K., Svarovskaia, E.S., and Pathak, V.K. (2000). Role of Murine Leukemia Virus Reverse Transcriptase Deoxyribonucleoside Triphosphate-Binding Site in Retroviral Replication and In Vivo Fidelity. J. Virol. *74*, 10349–10358. https://doi.org/10.1128/JVI.74.22.10349-10358.2000.Updated.

Heid, C.A., Stevens, J., Livak, K.J., and Williams, P.M. (1996). Real time quantitative PCR. Genome Res. *6*, 986–994. https://doi.org/10.1016/b978-012372185-3/50024-9.

Hershey, A.D., and Chase, M. (1952). INDEPENDENT FUNCTIONS OF VIRAL PROTEIN AND NUCLEIC ACID IN GROWTH OF BACTERIOPHAGE. J. Gen. Physiol. 39–56. .

Higuchi, R., Dollinger, G., Sean Walsh, P., and Griffith, R. (1992). Simultaneous amplification

and detection of specific DNA sequences. Bio/Technology *10*, 413–417. https://doi.org/10.1038/nbt0492-413.

Holland, P.M., Abramson, R.D., Watson, R., and Gelfand, D.H. (1991). Detection of specific polymerase chain reaction product by utilizing the 5′ → 3′ exonuclease activity of Thermus aquaticus DNA polymerase. Proc. Natl. Acad. Sci. U. S. A. *88*, 7276–7280. https://doi.org/10.1073/pnas.88.16.7276.

Huang, H., Chopra, R., Verdine, G.L., and Harrison, S.C. (1998). Structure of a Covalently Trapped Catalytic Complex of HIV-1 Reverse Transcriptase : Implications for Drug Resistance Published by : American Association for the Advancement of Science Stable URL : http://www.jstor.org/stable/2896852 Linked references are. *282*, 1669–1675. .

Jackson, D.A., Symons, R.H., and Berg, P. (1972). Biochemical method for inserting new genetic information into DNA of Simian Virus 40: circular SV40 DNA molecules containing lambda phage genes and the galactose operon of Escherichia coli. Proc. Natl. Acad. Sci. U. S. A. *69*, 2904–2909. https://doi.org/10.1073/pnas.69.10.2904.

Jacobo-Molina, A., and Arnold, E. (1991). HIV Reverse Transcriptase Structure-Function Relationships. Biochemistry *30*, 6351–6361. https://doi.org/10.1021/bi00240a001.

Jaeger, J., Restle, T., and Steitz, T.A. (1998). The structure of HIV-1 reverse transcriptase complexed with an RNA pseudoknot inhibitor. EMBO J. *17*, 4535–4542. https://doi.org/10.1093/emboj/17.15.4535.

Jarvis, T.C., and Kirkegaard, K. (1992). Poliovirus RNA recombination: Mechanistic studies in the absence of selection. EMBO J. *11*, 3135–3145. https://doi.org/10.1002/j.1460-2075.1992.tb05386.x.

Karst, S.M., Ziels, R.M., Kirkegaard, R.H., Sørensen, E.A., McDonald, D., Zhu, Q., Knight, R., and Albertsen, M. (2021). High-accuracy long-read amplicon sequences using unique molecular identifiers with Nanopore or PacBio sequencing. Nat. Methods *18*, 165–169. https://doi.org/10.1038/s41592-020-01041-y.

Kohlstaedt LA, Wang J, Friedman JM, Rice PA, S.T. 1992. complexed with an inhibitor. (1992). Crystal structure at 3.5 A resolution of HIV-1 reverse transcriptase. Science (80-.). *256*, 1783–1790. .

Koren, S., Walenz, B.P., Berlin, K., Miller, J.R., Bergman, N.H., and Phillippy, A.M. (2017). Canu: Scalable and accurate long-read assembly via adaptive κ-mer weighting and repeat separation. Genome Res. *27*, 722–736. https://doi.org/10.1101/gr.215087.116.

Krehenwinkel, H., Pomerantz, A., Henderson, J.B., Kennedy, S.R., Lim, J.Y., Swamy, V., Shoobridge, J.D., Graham, N., Patel, N.H., Gillespie, R.G., et al. (2019). Nanopore sequencing of long ribosomal DNA amplicons enables portable and simple biodiversity assessments with high phylogenetic resolution across broad taxonomic scale. Gigascience *8*, 1–16. https://doi.org/10.1093/gigascience/giz006.

Larder, B.A., Purifoy, D.J.M., Powell, K.L., and Darby, G. (1987). Site-specific mutagenesis of AIDS virus reverse transcriptase. Nature *327*, 716–717. https://doi.org/10.1038/327716a0.

Lee, J.Y., Kong, M., Oh, J., Lim, J.S., Chung, S.H., Kim, J.M., Kim, J.S., Kim, K.H., Yoo, J.C., and Kwak, W. (2021). Comparative evaluation of Nanopore polishing tools for microbial genome assembly and polishing strategies for downstream analysis. Sci. Rep. *11*, 1–11. https://doi.org/10.1038/s41598-021-00178-w.

Leferink, N.G.H., Dunstan, M.S., Hollywood, K.A., Swainston, N., Currin, A., Jervis, A.J., Takano, E., and Scrutton, N.S. (2019). An automated pipeline for the screening of diverse monoterpene synthase libraries. Sci. Rep. *9*, 1–12. https://doi.org/10.1038/s41598-019-48452-2.

Leipe, D.D., Aravind, L., and Koonin, E. V. (1999). Did DNA replication evolve twice independently? Nucleic Acids Res. *27*, 3389–3401. https://doi.org/10.1093/nar/27.17.3389.

Lenstra, R. (2015). The graph, geometry and symmetries of the genetic code with hamming metric. Symmetry (Basel). *7*, 1211–1260. https://doi.org/10.3390/sym7031211.

Liu, W., and Saint, D.A. (2002). A new quantitative method of real time reverse transcription polymerase chain reaction assay based on simulation of polymerase chain reaction kinetics. Anal. Biochem. *302*, 52–59. https://doi.org/10.1006/abio.2001.5530.

Lovenberg, W., and Sobel, B.E. (1965). Rubredoxin: a new electron transfer protein from Clostridium pasteurianum. Proc. Natl. Acad. Sci. U. S. A. *54*, 193–199. https://doi.org/10.1073/pnas.54.1.193.

Lundberg, K.S., Shoemaker, D.D., Adams, M.W.W., Short, J.M., Mathur, E.J., and Sorge, J.A. (1991). High-fidelity amplification using a thermostable DNA polymerase isolated from PyrOCOCCUS furiosus. Gene *108*, 1–6. https://doi.org/10.11918/j.issn.0367-6234.201705067.

Malik, O., Khamis, H., Rudnizky, S., and Kaplan, A. (2017). The mechano-chemistry of a monomeric reverse transcriptase. Nucleic Acids Res. *45*, 12954–12962. https://doi.org/10.1093/nar/gkx1168.

Menéndez-Arias, L. (2009). Mutation rates and intrinsic fidelity of retroviral reverse transcriptases. Viruses *1*, 1137–1165. https://doi.org/10.3390/v1031137.
Meselson, M., and Stahl, F.W. (1958). THE REPLICATION OF DNA IN ESCHERICHIA COLI*. Proc Natl Acad Sci U S A *44*, 671–682. https://doi.org/10.1007/s00265-017-2394-1.
Mohr, S., Ghanem, E., Smith, W., Sheeter, D., Qin, Y., King, O., Polioudakis, D., Iyer, V.R., Hunicke-Smith, S., Swamy, S., et al. (2013). Thermostable group II intron reverse transcriptase fusion proteins and their use in cDNA synthesis and next-generation RNA sequencing. Rna *19*, 958–970. https://doi.org/10.1261/rna.039743.113.
Moore, G.L., and Maranas, C.D. (2000). Modeling DNA mutation and recombination for directed evolution experiments. J. Theor. Biol. *205*, 483–503. https://doi.org/10.1006/jtbi.2000.2082.
Murphy, L.D., Herzog, C.E., Rudick, J.B., Fojo, A.T., and Bates, S.E. (1990). Use of the Polymerase Chain Reaction in the Quantitation of mdr-1 Gene Expression. Biochemistry *29*, 10351–10356. https://doi.org/10.1021/bi00497a009.
New England Biolabs (2019A) NEBNext® ultraTM RNA library prep kit for Illumina®, NEB. Available at: https://www.neb.com/en-gb/products/e7530-nebnext-ultra-rna-library-prep-kit-for-illumina (Accessed: 23 August 2022).
New England Biolabs. (2019B) Luna® Universal one-step RT-qPCR Kit, NEB. Available at: https://www.neb.com/en-gb/products/e3005-luna-universal-one-step-rt-qpcr-kit (Accessed: 23 August 2022).
Noonan, K.E., and Roninson, I.B. (1988). mRNA phenotyping by enzymatic amplification of randomly primed cDNA. Methods *16*, 10366. .
Ohtsubo, Y., Nagata, Y., and Tsuda, M. (2017). Compounds that enhance the tailing activity of Moloney murine leukemia virus reverse transcriptase. Sci. Rep. *7*, 1–6. https://doi.org/10.1038/s41598-017-04765-8.
Oscorbin, I.P., and Filipenko, M.L. (2021). M-MuLV reverse transcriptase: Selected properties and improved mutants. Comput. Struct. Biotechnol. J. *19*, 6315–6327. https://doi.org/10.1016/j.csbj.2021.11.030.
Painter, G.R., Wright, L.L., Hopkins, S., and Furman, P.A. (1990). UCLA Symposium and AIDS: Pathogenesis, Therapy, and Vaccine. J. Cell. Biochem.
Peirson, S.N., Butler, J.N., and Foster, R.G. (2003). Experimental validation of novel and conventional approaches to quantitative real-time PCR data analysis. Nucleic Acids Res. *31*.

https://doi.org/10.1093/nar/gng073.
Perez, O.D., and Nolan, G.P. (2001). Resistance is futile: Assimilation of cellular machinery by HIV-1. Immunity *15*, 687–690. https://doi.org/10.1016/S1074-7613(01)00238-2.
Pfaffl, M.W. (2001). A new mathematical model for relative quantification in real-time RT–PCR. Nucleic Acids Res. *29*, 2002–2007. https://doi.org/10.1111/j.1365-2966.2012.21196.x.
Pritchard, L., Corne, D., Kell, D., Rowland, J., and Winson, M. (2005). A general model of error-prone PCR. J. Theor. Biol. *234*, 497–509. https://doi.org/10.1016/j.jtbi.2004.12.005.
Renda, M.J., Rosenblatt, J.D., Klimatcheva, E., Demeter, L.M., Bambara, R.A., and Planelles, V. (2001). Mutation of the Methylated tRNA Lys Residue A58 Disrupts Reverse Transcription and Inhibits Replication of Human Immunodeficiency Virus Type 1. J. Virol. *75*, 9671–9678. https://doi.org/10.1189/jlb.2A0414-191R.
Rutledge, R.G., and Stewart, D. (2008). Critical evaluation of methods used to determine amplification efficiency refutes the exponential character of real-time PCR. BMC Mol. Biol. *9*, 1–12. https://doi.org/10.1186/1471-2199-9-96.
Rychlik, W. (1995). Selection of primers for polymerase chain reaction. Mol. Biotechnol. *3*, 129–134. https://doi.org/10.1007/BF02789108.
Saiki, R.K., Scharf, S., Faloona, F., Mullis, K.B., Horn, G.T., Erlich, H.A., and Arnheim, N. (1985). Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia. 1985. Science (80-.). *230*, 1350–1354. .
Saiki, R.K., Gelfand, D.H., Stoffel, S., Scharf, S.J., Horn, G.T., Mullis, K.B., Erlich, H.A., Saiki, R.K., Gelfand, D.H., Stoffel, S., et al. (1988). Primer-Directed Enzymatic Amplification of DNA with a Thermostable DNA Polymerase. Science (80-.). *239*, 487–491. .
Savolainen-Kopra, C., and Blomqvist, S. (2010). Mechanisms of genetic variation in polioviruses. Rev. Med. Virol. *20*, 358–371. .
Sawaya, M.R., Pelletier, H., Kumar, A., Wilson, S.H., and Kraut, J. (1994). Crystal structure of rat DNA polymerase β: Evidence for a common polymerase mechanism. Science (80-.). *264*, 1930–1935. https://doi.org/10.1126/science.7516581.
Schultz, S.J., Zhang, M., and Champoux, J.J. (2009). Preferred sequences within a defined cleavage window specify DNA 3′ end-directed cleavages by retroviral RNases H. J. Biol. Chem. *284*, 32225–32238. https://doi.org/10.1074/jbc.M109.043158.
Shannon, C.E. (1948). A Mathematical Theory of Communication. Bell Syst. Tech. J. *27*, 379-423,623-356. https://doi.org/10.1002/j.1538-7305.1948.tb00917.x.

Smith, H.O., and Welcox, K.W. (1970). A Restriction enzyme from Hemophilus influenzae. I. Purification and general properties. J. Mol. Biol. *51*, 379–391. https://doi.org/10.1016/0022-2836(70)90149-X.

Smith, D., Zhong, J., Matsuura, M., Lambowitz, A.M., and Belfort, M. (2005). Recruitment of host functions suggests a repair pathway for late steps in group II intron retrohoming. Genes Dev. *19*, 2477–2487. https://doi.org/10.1101/gad.1345105.

Spiess, A.N., Feig, C., and Ritz, C. (2008). Highly accurate sigmoidal fitting of real-time PCR data by introducing a parameter for asymmetry. BMC Bioinformatics *9*, 1–12. https://doi.org/10.1186/1471-2105-9-221.

Steitz, T.A. (1993). DNA- and RNA-dependent DNA polymerases. Curr. Opin. Struct. Biol. *3*, 31–38. .

Sun, F. (1995). The Polymerase Chain Reaction and Branching Processes. J. Comput. Biol. *2*, 63–86. https://doi.org/10.1089/cmb.1995.2.63.

Tarazona, P. (1992). Error thresholds for the molecular quasispecies as phase transitions: From simple landscapes to spin-glass models. Phys. Rev. A *45*. .

Temin, H.M., and Mizutani, S. (1970). RNA-dependent DNA Polymerase in Virions of Rous Sarcoma Virus. Nature *226*, 1211–1213. https://doi.org/10.1038/2261211a0.

Tournier, V., Topham, C.M., Gilles, A., David, B., Folgoas, C., Moya-Leclair, E., Kamionka, E., Desrousseaux, M.L., Texier, H., Gavalda, S., et al. (2020). An engineered PET depolymerase to break down and recycle plastic bottles. Nature *580*, 216–219. https://doi.org/10.1038/s41586-020-2149-4.

Tse, W.T., and Forget, B.G. (1990). Reverse transcription and direct amplification of cellular RNA transcripts by Taq polymerase. Gene *88*, 293–296. https://doi.org/10.1016/0378-1119(90)90047-U.

Vashishtha, A.K., and Konigsberg, W.H. (2016). Effect of Different Divalent Cations on the Kinetics and Fidelity of RB69 DNA Polymerase. Biochemistry *55*, 2661–2670. https://doi.org/10.1021/acs.biochem.5b01350.

Wang, D., Zhao, C., Cheng, R., and Sun, F. (2000). Estimation of the mutation rate during error-prone Polymerase Chain Reaction. J. Comput. Biol. *7*, 143–158. https://doi.org/10.1089/10665270050081423.

Watson, H.W., and Galton, F. (1875). On the Probability of the Extinction of Families. J. Anthropol. Inst. Gt. Britain Irel. *4*, 138–144. https://doi.org/10.2307/2841222.

Watson, J.D., and Crick, F.H.C. (1953). Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. Nature *171*, 737–738. https://doi.org/10.1038/171737a0.

Watts, J.M., Dang, K.K., Gorelick, R.J., Leonard, C.W., Bess, J.W., Swanstrom, R., Burch, C.L., and Weeks, K.M. (2009). Architecture and secondary structure of an entire HIV-1 RNA genome. Nature *460*, 711–716. https://doi.org/10.1038/nature08237.

Weiss, G., and Von Haeseler, A. (1997). A coalescent approach to the polymerase chain reaction. Nucleic Acids Res. *25*, 3082–3087. https://doi.org/10.1093/nar/25.15.3082.

Whiting, S.H., and Champoux, J.J. (1994). Strand displacement synthesis capability of Moloney murine leukemia virus reverse transcriptase. J. Virol. *68*, 4747–4758. .

Whiting, S.H., and Champoux, J.J. (1998). Properties of strand displacement synthesis by Moloney murine leukemia virus reverse transcriptase: Mechanistic implications. J. Mol. Biol. *278*, 559–577. https://doi.org/10.1006/jmbi.1998.1720.

Wu, J.W., Patterson-Lomba, O., Novitsky, V., and Pagano, M. (2015). A generalized entropy measure of within-host viral diversity for identifying recent HIV-1 infections. Med. (United States) *94*, e1865. https://doi.org/10.1097/MD.0000000000001865.

Zajac, P., Islam, S., Hochgerner, H., Lönnerberg, P., and Linnarsson, S. (2013). Base preferences in non-templated nucleotide incorporation by MMLV-derived reverse transcriptases. PLoS One *8*, 1–13. https://doi.org/10.1371/journal.pone.0085270.

Zhan, X., and Crouch, R.J. (1997). The isolated RNase H domain of murine leukemia virus reverse transeriptase: Retention of activity with concomitant loss of specificity. J. Biol. Chem. *272*, 22023–22029. https://doi.org/10.1074/jbc.272.35.22023.

Zhu, Y.Y., Machleder, E.M., Chenchik, A., Li, R., and Siebert, P.D. (2001). Reverse transcriptase template switching: A SMART^{TM} approach for full-length cDNA library construction. Biotechniques *30*, 892–897. https://doi.org/10.2144/01304pf02.

Editorial Nature, (1970). The central dogma reversed. Nature *227*, 1198–1199. .

https://nanoporetech.com/platform/accuracy, (2022). Oxford Nanopore accuracy.

Appendix 1.1: list of primers and targets used in the course of this work

Primer name	Target	Sequence	Uses
LacZα F	LacZα (pUC19)	CCTCTGACACATGCAGCTCC	PCR, EP-PCR, sequencing
LacZα R	LacZα (pUC19)	GGCAGTGAGCGCAACGCAAT	PCR, EP-PCR, sequencing
Rub Forward	NiFe gene	CCCCTCTAGAAATAATTTTGTTTAACT	PCR, EP-PCR
Rub Reverse	NiFe gene	CTCGAATTCGGATCCTCCCTCA	PCR, EP-PCR
T7 Forward	pET28a	CGAGATCTCGATCCCGCGAA	PCR, EP-PCR
T7 Reverse	pET28a	AGTTATTGCTCAGCGGTGGC	PCR, EP-PCR
T7 Forward 2	pET28a	CCGGCGTAGAGGATCGAGAT	PCR, EP-PCR, sequencing
T7 Reverse 2	pET28a	AGAGGCCCCAAGGGGTTATG	PCR, EP-PCR, sequencing
SDM STOP remove F1	RTase gene	TTCCCGCTTAATTAATggAGCGGCCGCTCGAGAGC	SDM removing STOP codon
SDM STOP remove R1	RTase gene	AGCTCTCGAGCGGCCGCTccATTAATTAAGCGGGA	SDM removing STOP codon
RT Mid F1	RTase gene	GTCGCCTCTGGATCCCgGGG	Sequencing
RT Mid R1	RTase gene	GATCCAGAGGCGACAGAAGC	Sequencing
pET28a Reverse UMI	pET28a	CAAGCAGAAGACGGCATACGAGATNNNYRNNNYRNNNYRNNNAGAGGCCCCAAGGGGTTATG	Sequencing
pET28a Forward UMI	pET28a	AATGATACGGCGACCACCGAGATCNNNYRNNNYRNNNYRNNNCCGGCGTAGAGGATCGAGAT	Sequencing
Forward UMI amplify	pET28a/UMI tagged sequence	AATGATACGGCGACCACCGAGATC	Sequencing
Reverse UMI amplify	pET28a/UMI tagged sequence	CAAGCAGAAGACGGCATACGAGAT	Sequencing
Act1 Forward	Act1 RNA	TAA CA ATG GAT TCT GG GTT GC	RT-PCR, RT-qPCR
Act1 Reverse	Act1 RNA	AAA CGT AGA AGG CTG GAA CG	RT-PCR, RT-qPCR

Snc1 F-DNA	Snc1 DNA/unspliced RNA	GTC ATC TAC TCC CTT TGA CC	RT-PCR, RT-qPCR
Snc1 F2-splice	Snc1 spliced RNA	CGG AAC TAC AAG CTG AAA TTG	RT-PCR, RT-qPCR
Snc1 reverse	Snc1 DNA/RNA	GGA CGA TGA TTA CAA CAA GC	RT-PCR, RT-qPCR
18S rRNA ITS1	Yeast 18S rRNA/DNA	TCC GTA GGT GAA CCT GCG G	RT-PCR, RT-qPCR
18S rRNA ITS4	Yeast 18S rRNA/DNA	TCC TCC GCT TAT TGA TAT GC	RT-PCR, RT-qPCR
16S rRNA 518R	Bacterial 16S rRNA/DNA	GTA TTA CCG CGG CTG CTG G	RT-PCR, RT-qPCR
16S rRNA 8F	Bacterial 16S rRNA/DNA	AGA GTT TGA TCC TGG CTC AG	RT-PCR, RT-qPCR

Appendix 1.2: List of barcoding primers used in course of this work

SampleID	FwIndex	FwPrimer	RvIndex	RvPrimer
OxNpBar_01	CACAAAGACACCGACAACTTTCTT	CCGGCGTAGAGGATCGAGAT	AAGAAAGTTGTCGGTGTCTTTGTG	AGAGGCCCCAAGGGGTTATG
OxNpBar_02	ACAGACGACTACAAACGGAATCGA	CCGGCGTAGAGGATCGAGAT	TCGATTCCGTTTGTAGTCGTCTGT	AGAGGCCCCAAGGGGTTATG
OxNpBar_03	CCTGGTAACTGGGACACAAGACTC	CCGGCGTAGAGGATCGAGAT	GAGTCTTGTGTCCCAGTTACCAGG	AGAGGCCCCAAGGGGTTATG
OxNpBar_04	TAGGGAAACACGATAGAATCCGAA	CCGGCGTAGAGGATCGAGAT	TTCGGATTCTATCGTGTTTCCCTA	AGAGGCCCCAAGGGGTTATG
OxNpBar_05	AAGGTTACACAAACCCTGGACAAG	CCGGCGTAGAGGATCGAGAT	CTTGTCCAGGGTTTGTGTAACCTT	AGAGGCCCCAAGGGGTTATG
OxNpBar_06	GACTACTTTCTGCCTTTGCGAGAA	CCGGCGTAGAGGATCGAGAT	TTCTCGCAAAGGCAGAAAGTAGTC	AGAGGCCCCAAGGGGTTATG
OxNpBar_07	AAGGATTCATTCCCACGGTAACAC	CCGGCGTAGAGGATCGAGAT	GTGTTACCGTGGGAATGAATCCTT	AGAGGCCCCAAGGGGTTATG
OxNpBar_08	ACGTAACTTGGTTTGTTCCCTGAA	CCGGCGTAGAGGATCGAGAT	TTCAGGGAACAAACCAAGTTACGT	AGAGGCCCCAAGGGGTTATG
OxNpBar_09	AACCAAGACTCGCTGTGCCTAGTT	CCGGCGTAGAGGATCGAGAT	AACTAGGCACAGCGAGTCTTGGTT	AGAGGCCCCAAGGGGTTATG
OxNpBar_10	GAGAGGACAAAGGTTTCAACGCTT	CCGGCGTAGAGGATCGAGAT	AAGCGTTGAAACCTTTGTCCTCTC	AGAGGCCCCAAGGGGTTATG
OxNpBar_11	TCCATTCCCTCCGATAGATGAAAC	CCGGCGTAGAGGATCGAGAT	GTTTCATCTATCGGAGGGAATGGA	AGAGGCCCCAAGGGGTTATG
OxNpBar_12	TCCGATTCTGCTTCTTTCTACCTG	CCGGCGTAGAGGATCGAGAT	CAGGTAGAAAGAAGCAGAATCGGA	AGAGGCCCCAAGGGGTTATG
OxNpBar_13	AGAACGACTTCCATACTCGTGTGA	CCGGCGTAGAGGATCGAGAT	TCACACGAGTATGGAAGTCGTTCT	AGAGGCCCCAAGGGGTTATG
OxNpBar_14	AACGAGTCTCTTGGGACCCATAGA	CCGGCGTAGAGGATCGAGAT	TCTATGGGTCCCAAGAGACTCGTT	AGAGGCCCCAAGGGGTTATG
OxNpBar_15	AGGTCTACCTCGCTAACACCACTG	CCGGCGTAGAGGATCGAGAT	CAGTGGTGTTAGCGAGGTAGACCT	AGAGGCCCCAAGGGGTTATG
OxNpBar_16	CGTCAACTGACAGTGGTTCGTACT	CCGGCGTAGAGGATCGAGAT	AGTACGAACCACTGTCAGTTGACG	AGAGGCCCCAAGGGGTTATG
OxNpBar_17	ACCCTCCAGGAAAGTACCTCTGAT	CCGGCGTAGAGGATCGAGAT	ATCAGAGGTACTTTCCTGGAGGGT	AGAGGCCCCAAGGGGTTATG
OxNpBar_18	CCAAACCCAACAACCTAGATAGGC	CCGGCGTAGAGGATCGAGAT	GCCTATCTAGGTTGTTGGGTTTGG	AGAGGCCCCAAGGGGTTATG
OxNpBar_19	GTTCCTCGTGCAGTGTCAAGAGAT	CCGGCGTAGAGGATCGAGAT	ATCTCTTGACACTGCACGAGGAAC	AGAGGCCCCAAGGGGTTATG
OxNpBar_20	TTGCGTCCTGTTACGAGAACTCAT	CCGGCGTAGAGGATCGAGAT	ATGAGTTCTCGTAACAGGACGCAA	AGAGGCCCCAAGGGGTTATG
OxNpBar_21	GAGCCTCTCATTGTCCGTTCTCTA	CCGGCGTAGAGGATCGAGAT	TAGAGAACGGACAATGAGAGGCTC	AGAGGCCCCAAGGGGTTATG
OxNpBar_22	ACCACTGCCATGTATCAAAGTACG	CCGGCGTAGAGGATCGAGAT	CGTACTTTGATACATGGCAGTGGT	AGAGGCCCCAAGGGGTTATG
OxNpBar_23	CTTACTACCCAGTGAACCTCCTCG	CCGGCGTAGAGGATCGAGAT	CGAGGAGGTTCACTGGGTAGTAAG	AGAGGCCCCAAGGGGTTATG
OxNpBar_24	GCATAGTTCTGCATGATGGGTTAG	CCGGCGTAGAGGATCGAGAT	CTAACCCATCATGCAGAACTATGC	AGAGGCCCCAAGGGGTTATG
OxNpBar_25	GTAAGTTGGGTATGCAACGCAATG	CCGGCGTAGAGGATCGAGAT	CATTGCGTTGCATACCCAACTTAC	AGAGGCCCCAAGGGGTTATG
OxNpBar_26	CATACAGCGACTACGCATTCTCAT	CCGGCGTAGAGGATCGAGAT	ATGAGAATGCGTAGTCGCTGTATG	AGAGGCCCCAAGGGGTTATG
OxNpBar_27	CGACGGTTAGATTCACCTCTTACA	CCGGCGTAGAGGATCGAGAT	TGTAAGAGGTGAATCTAACCGTCG	AGAGGCCCCAAGGGGTTATG
OxNpBar_28	TGAAACCTAAGAAGGCACCGTATC	CCGGCGTAGAGGATCGAGAT	GATACGGTGCCTTCTTAGGTTTCA	AGAGGCCCCAAGGGGTTATG
OxNpBar_29	CTAGACACCTTGGGTTGACAGACC	CCGGCGTAGAGGATCGAGAT	GGTCTGTCAACCCAAGGTGTCTAG	AGAGGCCCCAAGGGGTTATG
OxNpBar_30	TCAGTGAGGATCTACTTCGACCCA	CCGGCGTAGAGGATCGAGAT	TGGGTCGAAGTAGATCCTCACTGA	AGAGGCCCCAAGGGGTTATG
OxNpBar_31	TGCGTACAGCAATCAGTTACATTG	CCGGCGTAGAGGATCGAGAT	CAATGTAACTGATTGCTGTACGCA	AGAGGCCCCAAGGGGTTATG
OxNpBar_32	CCAGTAGAAGTCCGACAACGTCAT	CCGGCGTAGAGGATCGAGAT	ATGACGTTGTCGGACTTCTACTGG	AGAGGCCCCAAGGGGTTATG
OxNpBar_33	CAGACTTGGTACGGTTGGGTAACT	CCGGCGTAGAGGATCGAGAT	AGTTACCCAACCGTACCAAGTCTG	AGAGGCCCCAAGGGGTTATG
OxNpBar_34	GGACGAAGAACTCAAGTCAAAGGC	CCGGCGTAGAGGATCGAGAT	GCCTTTGACTTGAGTTCTTCGTCC	AGAGGCCCCAAGGGGTTATG
OxNpBar_35	CTACTTACGAAGCTGAGGGACTGC	CCGGCGTAGAGGATCGAGAT	GCAGTCCCTCAGCTTCGTAAGTAG	AGAGGCCCCAAGGGGTTATG
OxNpBar_36	ATGTCCCAGTTAGAGGAGGAAACA	CCGGCGTAGAGGATCGAGAT	TGTTTCCTCCTCTAACTGGGACAT	AGAGGCCCCAAGGGGTTATG
OxNpBar_37	GCTTGCGATTGATGCTTAGTATCA	CCGGCGTAGAGGATCGAGAT	TGATACTAAGCATCAATCGCAAGC	AGAGGCCCCAAGGGGTTATG
OxNpBar_38	ACCACAGGAGGACGATACAGAGAA	CCGGCGTAGAGGATCGAGAT	TTCTCTGTATCGTCCTCCTGTGGT	AGAGGCCCCAAGGGGTTATG
OxNpBar_39	CCACAGTGTCAACTAGAGCCTCTC	CCGGCGTAGAGGATCGAGAT	GAGAGGCTCTAGTTGACACTGTGG	AGAGGCCCCAAGGGGTTATG
OxNpBar_40	TAGTTTGGATGACCAAGGATAGCC	CCGGCGTAGAGGATCGAGAT	GGCTATCCTTGGTCATCCAAACTA	AGAGGCCCCAAGGGGTTATG
OxNpBar_41	GGAGTTCGTCCAGAGAAGTACACG	CCGGCGTAGAGGATCGAGAT	CGTGTACTTCTCTGGACGAACTCC	AGAGGCCCCAAGGGGTTATG
OxNpBar_42	CTACGTGTAAGGCATACCTGCCAG	CCGGCGTAGAGGATCGAGAT	CTGGCAGGTATGCCTTACACGTAG	AGAGGCCCCAAGGGGTTATG

OxNpBar_43	CTTTCGTTGTTGACTCGACGGTAG	CCGGCGTAGAGGATCGAGAT	CTACCGTCGAGTCAACAACGAAAG	AGAGGCCCCAAGGGGTTATG
OxNpBar_44	AGTAGAAAGGGTTCCTTCCCACTC	CCGGCGTAGAGGATCGAGAT	GAGTGGGAAGGAACCCTTTCTACT	AGAGGCCCCAAGGGGTTATG
OxNpBar_45	GATCCAACAGAGATGCCTTCAGTG	CCGGCGTAGAGGATCGAGAT	CACTGAAGGCATCTCTGTTGGATC	AGAGGCCCCAAGGGGTTATG
OxNpBar_46	GCTGTGTTCCACTTCATTCTCCTG	CCGGCGTAGAGGATCGAGAT	CAGGAGAATGAAGTGGAACACAGC	AGAGGCCCCAAGGGGTTATG
OxNpBar_47	GTGCAACTTTCCCACAGGTAGTTC	CCGGCGTAGAGGATCGAGAT	GAACTACCTGTGGGAAAGTTGCAC	AGAGGCCCCAAGGGGTTATG
OxNpBar_48	CATCTGGAACGTGGTACACCTGTA	CCGGCGTAGAGGATCGAGAT	TACAGGTGTACCACGTTCCAGATG	AGAGGCCCCAAGGGGTTATG
OxNpBar_49	ACTGGTGCAGCTTTGAACATCTAG	CCGGCGTAGAGGATCGAGAT	CTAGATGTTCAAAGCTGCACCAGT	AGAGGCCCCAAGGGGTTATG
OxNpBar_50	ATGGACTTTGGTAACTTCCTGCGT	CCGGCGTAGAGGATCGAGAT	ACGCAGGAAGTTACCAAAGTCCAT	AGAGGCCCCAAGGGGTTATG
OxNpBar_51	GTTGAATGAGCCTACTGGGTCCTC	CCGGCGTAGAGGATCGAGAT	GAGGACCCAGTAGGCTCATTCAAC	AGAGGCCCCAAGGGGTTATG
OxNpBar_52	TGAGAGACAAGATTGTTCGTGGAC	CCGGCGTAGAGGATCGAGAT	GTCCACGAACAATCTTGTCTCTCA	AGAGGCCCCAAGGGGTTATG
OxNpBar_53	AGATTCAGACCGTCTCATGCAAAG	CCGGCGTAGAGGATCGAGAT	CTTTGCATGAGACGGTCTGAATCT	AGAGGCCCCAAGGGGTTATG
OxNpBar_54	CAAGAGCTTTGACTAAGGAGCATG	CCGGCGTAGAGGATCGAGAT	CATGCTCCTTAGTCAAAGCTCTTG	AGAGGCCCCAAGGGGTTATG
OxNpBar_55	TGGAAGATGAGACCCTGATCTACG	CCGGCGTAGAGGATCGAGAT	CGTAGATCAGGGTCTCATCTTCCA	AGAGGCCCCAAGGGGTTATG
OxNpBar_56	TCACTACTCAACAGGTGGCATGAA	CCGGCGTAGAGGATCGAGAT	TTCATGCCACCTGTTGAGTAGTGA	AGAGGCCCCAAGGGGTTATG
OxNpBar_57	GCTAGGTCAATCTCCTTCGGAAGT	CCGGCGTAGAGGATCGAGAT	ACTTCCGAAGGAGATTGACCTAGC	AGAGGCCCCAAGGGGTTATG
OxNpBar_58	CAGGTTACTCCTCCGTGAGTCTGA	CCGGCGTAGAGGATCGAGAT	TCAGACTCACGGAGGAGTAACCTG	AGAGGCCCCAAGGGGTTATG
OxNpBar_59	TCAATCAAGAAGGGAAAGCAAGGT	CCGGCGTAGAGGATCGAGAT	ACCTTGCTTTCCCTTCTTGATTGA	AGAGGCCCCAAGGGGTTATG
OxNpBar_60	CATGTTCAACCAAGGCTTCTATGG	CCGGCGTAGAGGATCGAGAT	CCATAGAAGCCTTGGTTGAACATG	AGAGGCCCCAAGGGGTTATG
OxNpBar_61	AGAGGGTACTATGTGCCTCAGCAC	CCGGCGTAGAGGATCGAGAT	GTGCTGAGGCACATAGTACCCTCT	AGAGGCCCCAAGGGGTTATG
OxNpBar_62	CACCCACACTTACTTCAGGACGTA	CCGGCGTAGAGGATCGAGAT	TACGTCCTGAAGTAAGTGTGGGTG	AGAGGCCCCAAGGGGTTATG
OxNpBar_63	TTCTGAAGTTCCTGGGTCTTGAAC	CCGGCGTAGAGGATCGAGAT	GTTCAAGACCCAGGAACTTCAGAA	AGAGGCCCCAAGGGGTTATG
OxNpBar_64	GACAGACACCGTTCATCGACTTTC	CCGGCGTAGAGGATCGAGAT	GAAAGTCGATGAACGGTGTCTGTC	AGAGGCCCCAAGGGGTTATG
OxNpBar_65	TTCTCAGTCTTCCTCCAGACAAGG	CCGGCGTAGAGGATCGAGAT	CCTTGTCTGGAGGAAGACTGAGAA	AGAGGCCCCAAGGGGTTATG
OxNpBar_66	CCGATCCTTGTGGCTTCTAACTTC	CCGGCGTAGAGGATCGAGAT	GAAGTTAGAAGCCACAAGGATCGG	AGAGGCCCCAAGGGGTTATG
OxNpBar_67	GTTTGTCATACTCGTGTGCTCACC	CCGGCGTAGAGGATCGAGAT	GGTGAGCACACGAGTATGACAAAC	AGAGGCCCCAAGGGGTTATG
OxNpBar_68	GAATCTAAGCAAACACGAAGGTGG	CCGGCGTAGAGGATCGAGAT	CCACCTTCGTGTTTGCTTAGATTC	AGAGGCCCCAAGGGGTTATG
OxNpBar_69	TACAGTCCGAGCCTCATGTGATCT	CCGGCGTAGAGGATCGAGAT	AGATCACATGAGGCTCGGACTGTA	AGAGGCCCCAAGGGGTTATG
OxNpBar_70	ACCGAGATCCTACGAATGGAGTGT	CCGGCGTAGAGGATCGAGAT	ACACTCCATTCGTAGGATCTCGGT	AGAGGCCCCAAGGGGTTATG
OxNpBar_71	CCTGGGAGCATCAGGTAGTAACAG	CCGGCGTAGAGGATCGAGAT	CTGTTACTACCTGATGCTCCCAGG	AGAGGCCCCAAGGGGTTATG
OxNpBar_72	TAGCTGACTGTCTTCCATACCGAC	CCGGCGTAGAGGATCGAGAT	GTCGGTATGGAAGACAGTCAGCTA	AGAGGCCCCAAGGGGTTATG
OxNpBar_73	AAGAAACAGGATGACAGAACCCTC	CCGGCGTAGAGGATCGAGAT	GAGGGTTCTGTCATCCTGTTTCTT	AGAGGCCCCAAGGGGTTATG
OxNpBar_74	TACAAGCATCCCAACACTTCCACT	CCGGCGTAGAGGATCGAGAT	AGTGGAAGTGTTGGGATGCTTGTA	AGAGGCCCCAAGGGGTTATG
OxNpBar_75	GACCATTGTGATGAACCCTGTTGT	CCGGCGTAGAGGATCGAGAT	ACAACAGGGTTCATCACAATGGTC	AGAGGCCCCAAGGGGTTATG
OxNpBar_76	ATGCTTGTTACATCAACCCTGGAC	CCGGCGTAGAGGATCGAGAT	GTCCAGGGTTGATGTAACAAGCAT	AGAGGCCCCAAGGGGTTATG
OxNpBar_77	CGACCTGTTTCTCAGGGATACAAC	CCGGCGTAGAGGATCGAGAT	GTTGTATCCCTGAGAAACAGGTCG	AGAGGCCCCAAGGGGTTATG
OxNpBar_78	AACAACCGAACCTTTGAATCAGAA	CCGGCGTAGAGGATCGAGAT	TTCTGATTCAAAGGTTCGGTTGTT	AGAGGCCCCAAGGGGTTATG
OxNpBar_79	TCTCGGAGATAGTTCTCACTGCTG	CCGGCGTAGAGGATCGAGAT	CAGCAGTGAGAACTATCTCCGAGA	AGAGGCCCCAAGGGGTTATG
OxNpBar_80	CGGATGAACATAGGATAGCGATTC	CCGGCGTAGAGGATCGAGAT	GAATCGCTATCCTATGTTCATCCG	AGAGGCCCCAAGGGGTTATG
OxNpBar_81	CCTCATCTTGTGAAGTTGTTTCGG	CCGGCGTAGAGGATCGAGAT	CCGAAACAACTTCACAAGATGAGG	AGAGGCCCCAAGGGGTTATG
OxNpBar_82	ACGGTATGTCGAGTTCCAGGACTA	CCGGCGTAGAGGATCGAGAT	TAGTCCTGGAACTCGACATACCGT	AGAGGCCCCAAGGGGTTATG
OxNpBar_83	TGGCTTGATCTAGGTAAGGTCGAA	CCGGCGTAGAGGATCGAGAT	TTCGACCTTACCTAGATCAAGCCA	AGAGGCCCCAAGGGGTTATG
OxNpBar_84	GTAGTGGACCTAGAACCTGTGCCA	CCGGCGTAGAGGATCGAGAT	TGGCACAGGTTCTAGGTCCACTAC	AGAGGCCCCAAGGGGTTATG
OxNpBar_85	AACGGAGGAGTTAGTTGGATGATC	CCGGCGTAGAGGATCGAGAT	GATCATCCAACTAACTCCTCCGTT	AGAGGCCCCAAGGGGTTATG
OxNpBar_86	AGGTGATCCCAACAAGCGTAAGTA	CCGGCGTAGAGGATCGAGAT	TACTTACGCTTGTTGGGATCACCT	AGAGGCCCCAAGGGGTTATG

OxNpBar_87	TACATGCTCCTGTTGTTAGGGAGG	CCGGCGTAGAGGATCGAGAT	CCTCCCTAACAACAGGAGCATGTA	AGAGGCCCCAAGGGGTTATG
OxNpBar_88	TCTTCTACTACCGATCCGAAGCAG	CCGGCGTAGAGGATCGAGAT	CTGCTTCGGATCGGTAGTAGAAGA	AGAGGCCCCAAGGGGTTATG
OxNpBar_89	ACAGCATCAATGTTTGGCTAGTTG	CCGGCGTAGAGGATCGAGAT	CAACTAGCCAAACATTGATGCTGT	AGAGGCCCCAAGGGGTTATG
OxNpBar_90	GATGTAGAGGGTACGGTTTGAGGC	CCGGCGTAGAGGATCGAGAT	GCCTCAAACCGTACCCTCTACATC	AGAGGCCCCAAGGGGTTATG
OxNpBar_91	GGCTCCATAGGAACTCACGCTACT	CCGGCGTAGAGGATCGAGAT	AGTAGCGTGAGTTCCTATGGAGCC	AGAGGCCCCAAGGGGTTATG
OxNpBar_92	TTGTGAGTGGAAAGATACAGGACC	CCGGCGTAGAGGATCGAGAT	GGTCCTGTATCTTTCCACTCACAA	AGAGGCCCCAAGGGGTTATG
OxNpBar_93	AGTTTCCATCACTTCAGACTTGGG	CCGGCGTAGAGGATCGAGAT	CCCAAGTCTGAAGTGATGGAAACT	AGAGGCCCCAAGGGGTTATG
OxNpBar_94	GATTGTCCTCAAACTGCCACCTAC	CCGGCGTAGAGGATCGAGAT	GTAGGTGGCAGTTTGAGGACAATC	AGAGGCCCCAAGGGGTTATG
OxNpBar_95	CCTGTCTGGAAGAAGAATGGACTT	CCGGCGTAGAGGATCGAGAT	AAGTCCATTCTTCTTCCAGACAGG	AGAGGCCCCAAGGGGTTATG
OxNpBar_96	CTGAACGGTCATAGAGTCCACCAT	CCGGCGTAGAGGATCGAGAT	ATGGTGGACTCTATGACCGTTCAG	AGAGGCCCCAAGGGGTTATG

Appendix 2: List of programs utilised in the course of this project

2.1 – InitialBasicmodel.py

```
import math, time
startDNA=int(input("What is the starting concentration of DNA?"))
n=int(input("How many cycles of PCR?"))
errate=float(input("What is the error rate of the polymerase?"))
i=0
wt=startDNA
mut=0
prevwt=0
prevmut=0
test = 1
print('| %-20s | %-20s | %-20s | %-20s |' % ('Cycle','Total DNA','wt','mut'))
print('=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=/=')
while i<=n:
  totDNA=startDNA*2**i
  print('| %-20d | %-20d | %-20f | %-20f |' % (i, totDNA, wt, mut))
  test=totDNA-(wt+mut)
  prevwt=wt
  prevmut=mut
  wt+=prevwt*(1-errate)+0.25*prevmut*errate
  mut+=prevwt*errate+prevmut*(1-0.25*errate)
  i+=1
totwt=prevwt/totDNA
totmut=prevmut/totDNA
print (totwt, totmut)

time.sleep(2)
```

2.2 BasicStrandModel.py

```
from matplotlib import pyplot as plt
import numpy as np
cycles=25
mutrate=0.4

i=0
muts=[]
while i<1000:
    muts.append(0)
    i+=1
muts[0]=2
i=0
while i<cycles:
    j=len(muts)-1
    while j>=0:     #j starts at end as active cell and works back towards 0
        k=len(muts)-1-j
        totmutDNA=0
        while k>0: #k starts at (max - active) as mutating cell and works back towards j
            mutDNA=muts[j]*mutrate**k
            totmutDNA+=mutDNA
            muts[j+k]+=mutDNA
            k-=1
        muts[j]+=muts[j]-totmutDNA
        j-=1
    i+=1
print(muts)
print(sum(muts))
mutprob=[]
for i in muts:
    mutprob.append(i/sum(muts))
plt.figure()
plt.plot(np.linspace(0,1000,num=1000)[0:40],mutprob[0:40])

plt.xlabel("Number of Mutations")
plt.ylabel("Probability")
plt.title("Probability of mutation as iteratively simulated")
```

2.3.1 modellingtest1.py

```
import math, random, sys
i=0
DNA=[]
length=1000
while i<length:
 DNA.append(0)
 i+=1
DNAls=[DNA,DNA]
#err=float(sys.argv[1])
#cycle=int(sys.argv[2])
err=0.1
cycle=25
def randomise(inp):
 i=0
 while i<len(inp):
  z=random.uniform(0,1000/err)
  if 0.5<z<1.5:
   if inp[i]>0 and random.randint(0,2)==0:
    inp[i]=0
   else:
    inp[i]+=1
  #print(z,inp[i])
  i+=1
  #print(z)
 #print(type(inp))
 return(inp)


i=0
while i<cycle:
 new=[]
 test=[]
 for seq in DNAls:
  test=[]
  for j in seq:
   test.append(j)
  new.append(randomise(test))
  #new.append(seq)
  #print(new,seq,"\n",DNAls)
 for k in new:
  DNAls=[k]+DNAls
  #DNAls.append(k)
  #print(k)
 #print(len(DNAls))
 i+=1
 #print(i)

num=0
tot=0
ttot=0
for l in DNAls:
 tot+=sum(l)
 num+=1
 #print(tot,num,sum(l))
 for k in l:
  if k!=0:
   ttot+=1


avg=tot/(num)
tavg=ttot/num
#for i in DNAls:
 # print(i)
print(tavg)
```

2.3.2 IterateOverModel.bat

This program acted to call the modellingtest1.py program 100 times with the same parameters. The output of these calls were saved as .txt files, and were subsequently used to generate the graph.

2.3.3 SimulGraphM+SD.py

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D

def datasort(inp):
 data=[]
 k=0
 while k<10:
  data.append([])
  k+=1
 for i in inp:
  j=i.split("\t")
  k=0
  while k<len(j):
   data[k].append(j[k].strip())
   k+=1
 return(data)

def mean(lis):
   tot=0
   for i in lis:
      tot+=float(i)
   return(tot/len(lis))

def stdev(lis,mean):
   tot=0
   for i in lis:
      tot+=(float(i)-mean)**2
   return(np.sqrt(tot/(len(lis)-1)))


fig=plt.figure()
ax = fig.add_subplot(111, projection='3d')

tdata=open("simul0.1.txt").readlines()
data=datasort(tdata)
i=0
while i<len(data):
 imean=mean(data[i])
 istdev=stdev(data[i],imean)
 ax.scatter(0.1,i,mean(data[i]))
 ax.plot([0.1,0.1],[i,i],[imean-istdev,imean+istdev], marker="_")
 i+=1
 print(imean,istdev)


tdata=open("simul0.2.txt").readlines()
data=datasort(tdata)
i=0
while i<len(data):
 imean=mean(data[i])
 istdev=stdev(data[i],imean)
 ax.scatter(0.2,i,imean)
 ax.plot([0.2,0.2],[i,i],[imean-istdev,imean+istdev], marker="_")
 i+=1
 print(imean,istdev)


tdata=open("simul0.3.txt").readlines()
data=datasort(tdata)
i=0
while i<len(data):
```

```
  imean=mean(data[i])
  istdev=stdev(data[i],imean)
  ax.scatter(0.3,i,imean)
  ax.plot([0.3,0.3],[i,i],[imean-istdev,imean+istdev], marker="_")
  i+=1
  print(imean,istdev)

tdata=open("simul0.4.txt").readlines()
data=datasort(tdata)
i=0
while i<len(data):
  imean=mean(data[i])
  istdev=stdev(data[i],imean)
  ax.scatter(0.4,i,imean)
  ax.plot([0.4,0.4],[i,i],[imean-istdev,imean+istdev], marker="_")
  i+=1
  print(imean,istdev)


tdata=open("simul0.5.txt").readlines()
data=datasort(tdata)
i=0
while i<len(data):
  imean=mean(data[i])
  istdev=stdev(data[i],imean)
  ax.scatter(0.5,i,imean)
  ax.plot([0.5,0.5],[i,i],[imean-istdev,imean+istdev], marker="_")
  i+=1
  print(imean,istdev)

tdata=open("simul0.6.txt").readlines()
data=datasort(tdata)
i=0
while i<len(data):
  imean=mean(data[i])
  istdev=stdev(data[i],imean)
  ax.scatter(0.6,i,imean)
  ax.plot([0.6,0.6],[i,i],[imean-istdev,imean+istdev], marker="_")
  i+=1
  print(imean,istdev)

tdata=open("simul0.7.txt").readlines()
data=datasort(tdata)
i=0
while i<len(data):
  imean=mean(data[i])
  istdev=stdev(data[i],imean)
  ax.scatter(0.7,i,imean)
  ax.plot([0.7,0.7],[i,i],[imean-istdev,imean+istdev], marker="_")
  i+=1
  print(imean,istdev)

tdata=open("simul0.8.txt").readlines()
data=datasort(tdata)
i=0
while i<len(data):
  imean=mean(data[i])
  istdev=stdev(data[i],imean)
  ax.scatter(0.8,i,imean)
  ax.plot([0.8,0.8],[i,i],[imean-istdev,imean+istdev], marker="_")
  i+=1
  print(imean,istdev)

tdata=open("simul0.9.txt").readlines()
data=datasort(tdata)
i=0
while i<len(data):
  imean=mean(data[i])
  istdev=stdev(data[i],imean)
  ax.scatter(0.9,i,imean)
  ax.plot([0.9,0.9],[i,i],[imean-istdev,imean+istdev], marker="_")
```

```
 i+=1
 print(imean,istdev)

tdata=open("simul1.0.txt").readlines()
data=datasort(tdata)
i=0
while i<len(data):
 imean=mean(data[i])
 istdev=stdev(data[i],imean)
 ax.scatter(1.0,i,imean)
 ax.plot([1.0,1.0],[i,i],[imean-istdev,imean+istdev], marker="_")
 i+=1
 print(imean,istdev)

x=np.linspace(0,1,10)
y=np.linspace(0,10,10)
x,y=np.meshgrid(x,y)
z=(x/2)*y
#surf=ax.plot_surface(x,y,z, linewidth=0, antialiased=False, cmap=cm.coolwarm)

#ax.scatter(0,0,0,c="r",label="High")
#ax.scatter(0,0,0,c="b",label="Low")
ax.legend()
ax.set_xlabel('Polymerase Mutation Rate')
ax.set_ylabel('Number of Cycles')
ax.set_zlabel('Average Number of Mutations')

plt.show()
```

2.4 SimulatedEPPCR.py

```
# -*- coding: utf-8 -*-
"""
Created on Tue Aug 10 12:35:06 2021

@author: Jamie
"""
import sys, random
mutrate=700   #Mutrate determines the amount of mutations that will be introduced. 0<mutrate<1000000
indel=50
mutarray={"A":{"C":0.33,"T":0.33,"G":0.33,"-":0.01},
        "C":{"A":0.33,"T":0.33,"G":0.33,"-":0.01},
        "T":{"A":0.33,"C":0.33,"G":0.33,"-":0.01},
        "G":{"A":0.33,"C":0.33,"T":0.33,"-":0.01},
        "-":{"A":0.25,"C":0.25,"T":0.25,"G":0.25}}

def mutate(seq):
   global mutrate
   global mutarray
   out=""
   i=0
   while i<len(seq):
      if random.randint(0,1000000)<=mutrate:
         mutid=random.randint(0,100)
         j=0
         for prob in mutarray[seq[i]].keys():
            j+=mutarray[seq[i]][prob]*100
            if j>mutid:
               #print(j,mutid)
               out+=prob
               break
      else:
         out+=seq[i]
      i+=1
   #print(out)
   return(out)

fname=sys.argv[1]
f=open(fname)
cycle=int(fname.split("_")[1].split(".")[0])+1
outfile=fname.split("_")[0]+"_"+str(cycle)+".fasta"
newf=open(outfile,"a")
number=0
for line in f.readlines():
   #print(line)
   if line.startswith(">"):
      number+=1
      age=int(line[1:].split("_")[1][1:])+1
      name=line.split("_")[0]+"_a"+str(age)+"_c"+str(cycle)+"_n"+str(number)
      newseq=""
      newf.write(line.strip()+"\n")
   else:
      newseq=mutate(line.strip().upper())
      newf.write(line.strip()+"\n")
   if len(newseq)==0:
      continue
   else:
      newf.write(name.strip()+"\n"+newseq.upper().strip()+"\n")
```

2.5 StatEPPCR.py

```
# -*- coding: utf-8 -*-
"""
Created on Tue Aug 31 11:13:28 2021

@author: mbp18jcf
"""
import numpy as np
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
def summation(x):
   return(sum(x))

cycles=25
cycles+=1
mutrate=0.00009545
length=2077
muts=10
ampeff=0.5

data=np.zeros((cycles,muts))
bin=[]
for i in range(0,cycles):
   ibin1=(np.math.factorial(cycles)/(np.math.factorial(i)*np.math.factorial(cycles-i)))
   ibin2=(ampeff**i)*((1-ampeff)**(cycles-i))
   ibin=ibin1*ibin2
   bin.append(ibin)
   #print(ibin)
   pois=[]
   for j in range(0,muts):
      mutlen=length*i*mutrate
      ipois=(((mutlen**j)*np.e**(-mutlen))/np.math.factorial(j))*ibin
      pois.append(ipois)
      data[i][j]=ipois
      #print(ipois)
data=data
#print(data)
totmuts=[]
totmuts=np.apply_along_axis(summation,axis=0, arr=data)
#print(totmuts)


fig=plt.figure()
ax = Axes3D(fig)
ax.set_xlabel('Generation')
ax.set_ylabel("Mutations")
ax.set_zlabel("Frequency")
i=0
j=0
while i<cycles:
   j=0
   while j<muts:
      #print(i,j,data[i,j])
      ax.scatter(i,j,data[i,j])
      j+=1
   i+=1
#plt.plot(range(0,16),data.sum(axis=0))
#plt.plot(range(0,13),data.sum(axis=1))
#plt.show()

plt.figure()
plt.plot(np.linspace(0,muts-1,muts),totmuts)
```

```
plt.title("The probability of n mutations being introduced")
plt.xlabel("Number of mutations")
plt.ylabel("Probability")
i=0
mean=0
print("\n\n"+str(mutrate))
while i<len(totmuts):
   print(totmuts[i])
   mean+=totmuts[i]*(i+1)
   i+=1
print(mean)
stdev=0
i=0
while i<len(totmuts):
   stdev+=(i+1-mean)**2*totmuts[i]
   i+=1
stdev=np.sqrt(stdev)
print(stdev)
```

2.6 EPPCRAnalyse.py

```
# -*- coding: utf-8 -*-
"""
Created on Thu Aug 26 13:36:28 2021

@author: Jamie
"""
import sys
from BLAST import NWalgorithm
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

file=sys.argv[1]
wt=""
match=2
mismatch=-2
gap=-5
gapcontinue=-1
scoring=[match,mismatch,gap,gapcontinue]
cyclemut={}    #Saves the number of mutations on each product with cycle product was made in
agemut={}      #Saves the number of mutations on each product with associtated age of each product
totalmut=[]    #Saves the number of mutations on each product as a list
totaldata=[]   #Saves the age, cycle and number of mutations as a list of tuples

for line in open(file).readlines():
    if line[0]==">":
        lineinfo=line.split("_")[1:]
        age=int(lineinfo[0][1:])
        cycle=int(lineinfo[1][1:])
    elif cycle==0:
        wt=line.strip().upper()
    else:
        mut=0
        if len(line.strip())==len(wt):
            i=0
            while i<len(wt):
                if wt[i]!=line.strip()[i]:
                    mut+=1
                i+=1
                #print(i)
        else:
            s1,s2,z=NWalgorithm(wt,line.strip(),scoring,2)
            #print(s1,"\n####\n"+s2,z)
            i=0
            while i<len(s1):
                if s1[i]!=s2[i]:
                    mut+=1
                i+=1
        if cycle in cyclemut.keys():
            cyclemut[cycle].append(mut)
        else:
            cyclemut[cycle]=[mut]
        if age in agemut.keys():
            agemut[age].append(mut)
        else:
            agemut[age]=[mut]
        totalmut.append(mut)
        totaldata.append((age,cycle,mut))
print("###   AGE   ###")
for a in agemut.keys():
    average=sum(agemut[a])/len(agemut[a])
    print(a,average, sum(agemut[a]))
print("###   CYCLE   ###")
for a in cyclemut.keys():
    average=sum(cyclemut[a])/len(cyclemut[a])
    print(a,average,sum(cyclemut[a]))
print("###   TOTAL AVERAGE   ###\n"+str(sum(totalmut)/len(totalmut)),sum(totalmut))

plotdata=np.zeros((max(agemut.keys())+1,max(totalmut)+1))
```

```
#print(plotdata)
fig=plt.figure()
ax = Axes3D(fig)
ax.set_xlabel('Generation')
ax.set_ylabel("Mutations")
ax.set_zlabel("Frequency")
#ax2=plt.figure()
#ax2.add_axes()
ax3=plt.figure()
ax3.add_axes()
#ax3.set_xlabel("Generation")
for i in totaldata:
   plotdata[i[0],i[2]]+=1
print(plotdata)
probdata=plotdata/plotdata.sum()
print(probdata)
i=0
while i<len(plotdata):
   j=0
   while j<len(plotdata[i]):
      ax.scatter(i,j,plotdata[i,j])
      #plt.scatter(i,plotdata.sum(axis=1)[i])
      plt.xlabel("Generation")
      plt.ylabel("Frequency")
      #plt.scatter(j,plotdata.sum(axis=0)[j-1])
      j+=1
   i+=1
plt.plot(range(0,max(agemut.keys())+1),plotdata.sum(axis=1))
plt.plot(range(0,max(totalmut)+1),plotdata.sum(axis=0))
plt.show()
```

2.7 EPPCRLEastSpONLYMUT.py

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jul 18 12:04:34 2022

@author: james
"""
import numpy as np
from matplotlib import pyplot as plt
from scipy import optimize
#import scipy.optimize

def summation(x):
  return(sum(x))

def mutate(mutrates,*args,**kwargs):    ###Function to simulate EP-PCR
  mutrate=mutrates
  if "cycnum" in kwargs.keys():
    cycnum=int(kwargs["cycnum"])
  else:
    cycnum=25
  if "ampeff" in kwargs.keys():
    ampeff=float(kwargs["ampeff"])
  else:
    ampeff=0.5
  if "genelen" in kwargs.keys():
    genelen=kwargs["genelen"]
  else:
    genelen=437
  if "mutnum" in kwargs.keys():
    mutnum=kwargs["mutnum"]
  else:
    mutnum=10

  cycles=np.linspace(0,cycnum,cycnum+1)
  muts=np.linspace(0,mutnum,mutnum+1)
  binom=[]
  global test
  test=np.zeros([int(mutnum)+1,int(cycnum)+1])
  for cycle in cycles:
    binom.append((np.math.factorial(cycnum)/
          (np.math.factorial(cycle)*np.math.factorial(cycnum-cycle))
          )*((ampeff**cycle)*(1-ampeff)**(cycnum-cycle)))
  global iterations
  iterations=[x for x in binom]
  #return(iterations)
  #print(iterations)
  i=0
  while i<len(iterations):
    iterate=iterations[i]
   #print(iterate)
    for mut in muts:
      #print(mut,i,iterate)
      test[int(mut)][i]=((((mutrate*i*genelen)**mut)*np.exp(-mutrate*i*genelen)
            )/np.math.factorial(mut))*iterate
    i+=1
  #print(test)
  #return(test)
  return(np.apply_along_axis(summation, axis=1, arr=test))

def leastsqinput(mutrate):    #Function to input into leastsquares to minimise residuals
  return(data-mutate(mutrate,mutnum=10, genelen=437))
```

```
x0=0.001

###Rub Data
tencycdata=[0.392857,0.285714,0.178571,0,0.107143,0.0,0.035714,0.0,0,0,0]
twncycdata=[0.608696,0.173913,0.086957,0.043478,0.043478,0.0,0.0,0.0,0.0,0.0,0.0,0.043478]
twnfcycdata=[0.333333,0.416667,0.083333,0.16667,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0]

##10cycles
data=tencycdata
tenlsout=optimize.least_squares(leastsqinput,x0,jac="3-point",bounds=(0,0.5),loss="soft_l1")
print("10",tenlsout["x"])
print(tenlsout["cost"])
plt.plot(mutate(tenlsout["x"],mutnum=10))
plt.plot(data)
plt.title("Probability of number of mutation after 10 cycles of EP-PCR")
plt.xlabel("Number of mutations")
plt.ylabel("Probability")
plt.legend(["Simulated data","Sequenced data"])
plt.show()

##20cycles
data=twncycdata
twnlsout=optimize.least_squares(leastsqinput,x0,jac="3-point",bounds=(0,0.5),loss="soft_l1")
print("20",twnlsout["x"])
print(twnlsout["cost"])
plt.plot(mutate(twnlsout["x"],mutnum=10))
plt.plot(data)
plt.title("Probability of number of mutation after 20 cycles of EP-PCR")
plt.xlabel("Number of mutations")
plt.ylabel("Probability")
plt.legend(["Simulated data","Sequenced data"])
plt.show()

##25cycles
data=twnfcycdata
twnflsout=optimize.least_squares(leastsqinput,x0,jac="3-point",bounds=(0,0.5),loss="soft_l1")
print("25",twnflsout["x"])
print(twnflsout["cost"])
plt.plot(mutate(twnflsout["x"],mutnum=10))
plt.plot(data)
plt.title("Probability of number of mutation after 25 cycles of EP-PCR")
plt.xlabel("Number of mutations")
plt.ylabel("Probability")
plt.legend(["Simulated data","Sequenced data"])
plt.show()
```

2.8 EPPCRLeastSq+AMPEFF.py

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jul 18 12:04:34 2022

@author: james
"""

import numpy as np
from matplotlib import pyplot as plt
from scipy import optimize
#import scipy.optimize

def summation(x):
   return(sum(x))

def mutate(mutrates,*args,**kwargs):    ###Function to simulate EP-PCR
   mutrate=mutrates[0]
   ampeff=mutrates[1]
   if "cycnum" in kwargs.keys():
      cycnum=int(kwargs["cycnum"])
   else:
      cycnum=25
   # if "ampeff" in kwargs.keys():
   #    ampeff=float(kwargs["ampeff"])
   # else:
   #    ampeff=0.5
   if "genelen" in kwargs.keys():
      genelen=kwargs["genelen"]
   else:
      genelen=437
   if "mutnum" in kwargs.keys():
      mutnum=kwargs["mutnum"]
   else:
      mutnum=10

   cycles=np.linspace(0,cycnum,cycnum+1)
   muts=np.linspace(0,mutnum,mutnum+1)
   binom=[]
   global test
   test=np.zeros([int(mutnum)+1,int(cycnum)+1])
   for cycle in cycles:
      binom.append((np.math.factorial(cycnum)/
                (np.math.factorial(cycle)*np.math.factorial(cycnum-cycle))
                )*((ampeff**cycle)*(1-ampeff)**(cycnum-cycle)))
   global iterations
   iterations=[x for x in binom]
   #return(iterations)
   #print(iterations)
   i=0
   while i<len(iterations):
      iterate=iterations[i]
     #print(iterate)
      for mut in muts:
         #print(mut,i,iterate)
         test[int(mut)][i]=((((mutrate*i*genelen)**mut)*np.exp(-mutrate*i*genelen)
                     )/np.math.factorial(mut))*iterate
      i+=1
   #print(test)
   #return(test)
   return(np.apply_along_axis(summation, axis=1, arr=test))
```

```
def leastsqinput(mutrate):     #Function to input into leastsquares to minimise residuals
   return(data-mutate(mutrate,mutnum=10, genelen=437))



x0=[0.0001,0.5]

###Rub Data
tencycdata=[0.392857,0.285714,0.178571,0,0.107143,0.0,0.035714,0.0,0,0,0]
twncycdata=[0.608696,0.173913,0.086957,0.043478,0.043478,0.0,0.0,0.0,0.0,0.0,0.0,0.043478]
twnfcycdata=[0.333333,0.416667,0.083333,0.16667,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0]

##10cycles + ampeff
data=tencycdata
tenlsout=optimize.least_squares(leastsqinput,x0,jac="3-point",bounds=(0,0.5),loss="soft_l1")
print("10",tenlsout["x"])
print(tenlsout["cost"])
plt.plot(mutate([tenlsout["x"][0],tenlsout["x"][1]],mutnum=10))
plt.plot(data)
plt.title("Probability of number of mutation after 10 cycles of EP-PCR")
plt.xlabel("Number of mutations")
plt.ylabel("Probability")
plt.legend(["Simulated data","Sequenced data"])
plt.show()

##20cycles + ampeff
data=twncycdata
twnlsout=optimize.least_squares(leastsqinput,x0,jac="3-point",bounds=(0,0.5),loss="soft_l1")
print("20",twnlsout["x"])
print(twnlsout["cost"])
plt.plot(mutate([twnlsout["x"][0],twnlsout["x"][1]],mutnum=10))
plt.plot(data)
plt.title("Probability of number of mutation after 20 cycles of EP-PCR")
plt.xlabel("Number of mutations")
plt.ylabel("Probability")
plt.legend(["Simulated data","Sequenced data"])
plt.show()

##25cycles +ampeff
data=twnfcycdata
twnflsout=optimize.least_squares(leastsqinput,x0,jac="3-point",bounds=(0,0.5),loss="soft_l1")
print("25",twnflsout["x"])
print(twnflsout["cost"])
plt.plot(mutate([twnflsout["x"][0],twnflsout["x"][1]],mutnum=10))
plt.plot(data)
plt.title("Probability of number of mutation after 25 cycles of EP-PCR")
plt.xlabel("Number of mutations")
plt.ylabel("Probability")
plt.legend(["Simulated data","Sequenced data"])
plt.show()
```

2.9 ModelLSS

```
 # -*- coding: utf-8 -*-
"""
Created on Mon Mar  4 09:31:09 2024

@author: james
"""
from math import exp
from math import factorial as fact
from matplotlib import pyplot as plt
from scipy import optimize
import numpy as np

def summation(x):
    return(sum(x))

def WANG(x,cycles,length):
    allprob=[]
    mutrate=exp(-x[0])
    ampeff=x[1]
    for m in range(22):
        mprob=0
        for k in range(cycles+1):
            pk=0.25+0.75*((4/3)*mutrate-(1/3))**k
            prob1=(((fact(length)/(fact(m)*fact(length-m)))
                    *((1-pk)**m)*pk**(length-m))
                    *(((fact(cycles)/(fact(k)*fact(cycles-k)))
                    *ampeff**k))
                    /((1+ampeff)**cycles))
            mprob+=prob1
        allprob.append(mprob)
    return(np.array(allprob))

def PritchMc(x,cycle,S0,length):
    mutrate=x[0]
    ampeff=x[1]
    allznn=[]
    muts={}
    totznn=0
    for m in range(cycle):
        muts[m]=[]
    for n in range(cycle+1):
        ZNn=(fact(cycle)/(fact(n)*fact(cycle-n)))*(
            S0*ampeff**n)
        # print(n,ZNn)
        allznn.append(ZNn)
        totznn+=ZNn
        for m in range(n):
            probm=(fact(n)/(fact(m)*fact(n-m)))*(mutrate)**m*(1-mutrate)**(n-m)
            muts[m].append(ZNn*probm)
        # print(n,ZNn)
    allmuts=[]
    for i in muts.keys():
        allmuts.append(summation(muts[i]))
        if i>=21:
            break
    return(np.array(allmuts)/totznn)

def Moore(x,cycle,length):
    mutrate=x[0]
    ampeff=1
    allznn=[]
    muts={}
    totznn=0
    for m in range(cycle):
        muts[m]=[]
```

```
    for n in range(cycle+1):
        ZNn=(fact(cycle)/(fact(n)*fact(cycle-n)))*(
            2)
        # print(n,ZNn)
        allznn.append(ZNn)
        totznn+=ZNn
        for m in range(n):
            probm=(fact(n)/(fact(m)*fact(n-m)))*(mutrate)**m*(1-mutrate)**(n-m)
            muts[m].append(ZNn*probm)
        # print(n,ZNn)
    allmuts=[]
    for i in muts.keys():
        allmuts.append(summation(muts[i]))
        if i>=21:
            break
    return(np.array(allmuts)/totznn)


def ownmodel(x,cycle,length):    ###Function to simulate EP-PCR
    mutrate=x[0]
    ampeff=x[1]
    cycles=np.linspace(0,cycle,cycle+1)
    muts=np.linspace(0,21,21+1)
    binom=[]
    genelen=1349
    global test
    test=np.zeros([int(21)+1,int(cycle)+1])
    for n in cycles:
        binom.append((np.math.factorial(cycle)/
                      (np.math.factorial(n)*np.math.factorial(cycle-n))
                      )*((ampeff**n)*(1-ampeff)**(cycle-n)))
    global iterations
    iterations=[x for x in binom]
    #return(iterations)
    #print(iterations)
    i=0
    while i<len(iterations):
        iterate=iterations[i]
       #print(iterate)
        for mut in muts:
            #print(mut,i,iterate)
            test[int(mut)][i]=((((mutrate*i*genelen)**mut)*np.exp(-
mutrate*i*genelen)
                               )/np.math.factorial(mut))*iterate
        i+=1
    return(np.apply_along_axis(summation, axis=1, arr=test))

def WANGleastsqinput(mutrate):      #Function to input into leastsquares to
minimise residuals
    global data
    return(data-WANG(mutrate,35,1349))

def Pritchleastsqinput(mutrate):      #Function to input into leastsquares to
minimise residuals
    global data
    return(data-PritchMc(mutrate,35,2,1349))

def Ownleastsqinput(mutrate):
    global data
    return(data-ownmodel(mutrate,35,1349))

def MOOREleastsqinput(mutrate):
    global data
    return(data-Moore(mutrate,35,1349))

Taq=[0.004716981,0.009433962,0.004716981,0.023584906,0,0.009433962,0.033018868,0.10
3773585,0.075471698,0.089622642,0.08490566,0.075471698,0.108490566,0.08490566,0.051
```

```
886792,0.066037736,0.037735849,0.023584906,0.051886792,0.009433962,0.047169811,0.00
4716981]
PhoEP=[0.326923077,0.389423077,0.158653846,0.057692308,0.0625,0.004807692,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0]
data=PhoEP
x0=[0,0.1]

WANGlsout=optimize.least_squares(WANGleastsqinput,x0,jac="3-
point",bounds=(0,1),loss="soft_l1")
Pritchlsout=optimize.least_squares(Pritchleastsqinput,x0,jac="3-
point",bounds=(0,1),loss="soft_l1")
Ownlsout=optimize.least_squares(Ownleastsqinput,x0,jac="3-
point",bounds=(0,0.5),loss="soft_l1")
moorelsout=optimize.least_squares(MOOREleastsqinput,x0,jac="3-
point",bounds=(0,0.5),loss="soft_l1")

print("WANG",WANGlsout["x"],WANGlsout["cost"])
print("PRITCH",Pritchlsout["x"][0]/1349,Pritchlsout["x"][1],Pritchlsout["cost"])
print("OWN",Ownlsout["x"],Ownlsout["cost"])
print("Moore",moorelsout["x"][0]/1349,moorelsout["x"][1],moorelsout["cost"])



mutrate=WANGlsout["x"][0]
ampeff=WANGlsout["x"][1]
plt.plot(WANG([mutrate,ampeff],35,1349),"^--", label="Wang model")
plt.plot(data, label="PhoEP data")
plt.plot(PritchMc(Pritchlsout["x"],35,2,1349),"v--", label="Pritchard model")
plt.plot(ownmodel(Ownlsout["x"],35,1349),"--", label="Own model")
plt.plot(Moore(moorelsout["x"],35,1349),"-.",label="Moore model")
plt.ylabel("Proportion of sequences")
plt.xlabel("Number of mutations per 1349bp DNA sequence")
plt.title("Non-linear least square optimised models for PhoEP data")
plt.legend()
```

2.10 AnalyseDNAPairwise2.py

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jan  8 09:50:01 2024

@author: james
"""
import pandas as pd
from Bio import pairwise2 as pw
from matplotlib import pyplot as plt

file = "C:\\Users\\james\\OneDrive - entropix.co.uk\\Personal\\PhD\\CORRECTIONS\\NewData\\"
file="E:\\EntropixOneDrive\\OneDrive - entropix.co.uk\\Personal\\PhD\\CORRECTIONS\\NewData\\"
WT = open(file+"WTHOTPETase.fasta").readlines()[1]
series="B"
file = file+series+"_HOTPETaseORF.csv"
allseqs={}

datadf = pd.read_csv(file,header=None,names=["Clone","Seq"]).dropna()
mutations={}
nummuts=[]
for index in datadf.index:
    muttypes=[]
    if index==0:
        continue
    seq = datadf["Seq"][index]
    if len(seq)<1000:
        continue
    alignments = pw.align.globalms(WT,seq,1,-2,-5,-1)
    target=alignments[0][0]
    query=alignments[0][1]
    if seq in allseqs.keys():
        allseqs[seq].append(index)
    else:
        allseqs[seq]=[index]

    i=0
    mut=0
    while i<len(query):
        if target[i]!=query[i]:
            muttypes.append(target[i]+"->"+query[i])
            mut+=1
        i+=1
    #print(str(index)+":"+str(mut))
    if mut>100:
        print(str(index)+" : "+str(mut))
        # print(query)
        # print(target)
    else:
        nummuts.append(mut)
        print(index)
        for i in muttypes:
            print(i)
            if i not in mutations.keys():
                mutations[i]=1
            else:
                mutations[i]+=1
sortmuts=sorted(nummuts)

plt.rcParams['figure.figsize'] = [20,20]
plt.rcParams['font.size'] = 28
#plt.hist([x for x in nummuts], bins = max(sortmuts), rwidth = 0.5, align="left")
```

```
print(len(nummuts))
plt.hist([x for x in nummuts],bins=max(sortmuts), align="left", rwidth=0.5, weights=[((x+1)/(x+1))/len(nummuts) for x in
range(len(nummuts))])
plt.xlabel("Number of mutations")
plt.ylabel("Proportion of sequenced population")
if series=="B":
   plt.title("Number of mutations introduced after a 35 cycle EP-PCR using Taq/Mn\u00B2\u207A")
else:
   plt.title("Number of mutations introduced after a 35 cycle EP-PCR using PhoEP")

for i in mutations.keys():
   print(i,mutations[i])

numdict={}
for i in nummuts:
   if i in numdict.keys():
      numdict[i]+=1
   else:
      numdict[i]=1
print("")
for i in sorted(numdict.keys()):
   print(i,numdict[i])

for i in allseqs.keys():
   print(allseqs[i])
#print(datadf)
```

2.11 Shell scripts to run MiniBar and Canu on NiFe-PETase data

2.11.1 MinibarToCanu.sh

```
#!/bin/sh
folder="/home/james/Documents/Entropix/OxfordNanopore/HOTPETase/A091
-185/fastq_pass/"

barcodes="/home/james/Documents/Entropix/OxfordNanopore/Canu/96_T7_O
xNpbarcodes.txt"
newdir=$(echo "$folder"|awk -F/ '{print $(NF-2)}')
echo $newdir
mkdir $newdir
for file in "$folder"*.gz;
do
  cat "$file">>$newdir/allfile.fastq.gz
done

minibar.py -F "$barcodes" $newdir/allfile.fastq.gz
for barcode in ./*.fastq
do
  name=$(echo $barcode|awk -F/ '{print $NF}'|awk -F. '{print $1}')
  canu -p $name -d $name genomeSize=1.65k -nanopore $barcode
  mv $barcode $newdir
done
bash Compiler.sh
```

2.11.2 Compile.sh

```
#!/bin/sh
folder='/home/james/Documents/Entropix/OxfordNanopore/Canu'

for barcode in "$folder"/sample*/*contigs.fasta
do
  name=$(echo $barcode|awk -F/ '{print $NF}'|awk -F. '{print $1}')
  echo ">"$name >> allfiles.fasta
  tail -n +2 $barcode >>allfiles.fasta
done
```

2.12 EPPCRLeastSqRT

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jul 18 12:04:34 2022

@author: james
"""

import numpy as np
from matplotlib import pyplot as plt
from scipy import optimize
#import scipy.optimize

def summation(x):
    return(sum(x))

def mutate(mutrates,*args,**kwargs):    ###Function to simulate EP-PCR
    mutrate=mutrates[0]
    ampeff=mutrates[1]
    #ampeff=0.5
    if "cycnum" in kwargs.keys():
        cycnum=int(kwargs["cycnum"])
    else:
        cycnum=25
    # if "ampeff" in kwargs.keys():
    #     ampeff=float(kwargs["ampeff"])
    # else:
    #     ampeff=0.5
    #ampeff=0.5
    if "genelen" in kwargs.keys():
        genelen=kwargs["genelen"]
    else:
        genelen=2077
    if "mutnum" in kwargs.keys():
        mutnum=kwargs["mutnum"]
    else:
        mutnum=10

    cycles=np.linspace(0,cycnum,cycnum+1)
    muts=np.linspace(0,mutnum,mutnum+1)
    binom=[]
    global test
    test=np.zeros([int(mutnum)+1,int(cycnum)+1])
    for cycle in cycles:
        binom.append((np.math.factorial(cycnum)/
                      (np.math.factorial(cycle)*np.math.factorial(cycnum-cycle))
                      )*((ampeff**cycle)*(1-ampeff)**(cycnum-cycle)))
    global iterations
    iterations=[x for x in binom]
    #return(iterations)
    #print(iterations)
    i=0
    while i<len(iterations):
        iterate=iterations[i]
       #print(iterate)
        for mut in muts:
            #print(mut,i,iterate)
            test[int(mut)][i]=((((mutrate*i*genelen)**mut)*np.exp(-mutrate*i*genelen)
                               )/(np.math.factorial(mut))*iterate)
        i+=1
   # print(test)
    #return(test)
    return(np.apply_along_axis(summation, axis=1, arr=test))

def leastsqinput(mutrate):      #Function to input into leastsquares to minimise residuals
    return(data-mutate(mutrate,mutnum=8, genelen=2077))


###RT DATA###
x0=[0.00123,0.5]
#x0=[0.001]
intdata=[20,11,15,20,10,5,2,0,1]
data=[x/sum(intdata) for x in intdata]
#data=mutate(0.0002)
```

```
# while x0[0]<0.5:
leastsquaresout=optimize.least_squares(leastsqinput,x0,jac='3-
point',bounds=(0,0.5),loss="soft_l1")
# leastsqout=optimize.leastsq(leastsqinput,x0)
print(leastsquaresout)#["x"])
# print(leastsqout[0])
print(leastsquaresout["cost"])
# plt.plot(mutate([leastsquaresout["x"][0],leastsquaresout["x"][1]],mutnum=8))
# plt.plot(data)
# plt.show()

plt.plot(mutate([leastsquaresout["x"][0],leastsquaresout["x"][1]],mutnum=8))
plt.plot(data)
plt.xlabel("Number of mutations")
plt.ylabel("Proportion of sequences")
plt.title("Comparison of sequenced and simulated data using NLLS derived parameters")
plt.legend(["Simulated","Sequenced"])
plt.show()
```