STATISTICAL MODELING AND ANALYSIS OF THE POLYMERASE CHAIN REACTION

by

TERRI LYNN HENDERSON

(Under the direction of Dr. Anand Vidyashankar)

ABSTRACT

Polymerase chain reaction (PCR) is a popular method of detecting the presence of a target gene. A mathematical model is developed based upon the principals of branching processes and confidence intervals for various methods of determining a probability of reproduction are given. The model is tested using simulations based upon the stochastic process of PCR. The model proves to be a valid and useful method for determining the initial copy number of a target, regardless of the size of the copy number.

INDEX WORDS: Polymerase chain reaction, Mathematical model, Branching process.

STATISTICAL MODELING AND ANALYSIS OF THE POLYMERASE CHAIN REACTION

by

TERRI LYNN HENDERSON

B.S.A., The University of Georgia, 1999

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirement for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2002

STATISTICAL MODELING AND ANALYSIS OF THE POLYMERASE CHAIN REACTION

by

TERRI LYNN HENDERSON

Approved:

Major Professor: Anand Vidyashankar

Committee: Robert Taylor
Paul Schliekelman

Electronic Version Approved:

Gordhan L. Patel
Dean of the Graduate School
The University of Georgia
May 2002

DEDICATION

I want to dedicate this effort to my supportive husband Michael, for all he went through as I finished. His support and help got me through as nothing else could have. I greatly appreciate and thank the members of my committee, Dr. Bob Taylor and Dr. Paul Schliekelman, for their time and effort. Many great thanks go to my major professor, Dr. Anand Vidyashankar, who worried that I would not complete everything in time, and who directed me in the research. Finally, thanks be to Christ, who saved me, and kept me sane throughout.

TABLE OF CONTENTS

CHAPTER 1

POLYMERASE CHAIN REACTION

1.1 Genetic Background

Polymerase chain reaction (PCR) is a biotechnological method for amplifying, or increasing the number of, specific sequences of nucleic acids. Deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) are the two types of nucleic acids and are the genetic materials of all organisms. DNA is composed of sequences of 4 nucleotides, which are molecules composed of carbon, hydrogen, nitrogen, oxygen, and phosphorus atoms. Some of the carbon, hydrogen, and oxygen molecules of nucleotides make up a sugar for the nucleotides. The sugar for DNA is call deoxyribose. The four nucleotides (bases) in DNA are adenine (A), thymine (T), guanine (G), and cytosine (C). In a DNA molecule, a non-overlapping sequence of three bases comprise a codon, and sequences of codons make up genes. The structure of DNA is a double helix, which is composed of double-stranded lengths of nucleotides. One of the strands of DNA is known as the "sense" strand and the other the "antisense" strand. An adenine in the sense strand is always paired with a thymine in the antisense strand and vice versa, while guanine is paired with cytosine. DNA can range in length from a few hundred base pairs (a base pair, bp, is a sense-antisense pair of nucleotides) to millions of base pairs.

RNA is composed of the same nucleotides as DNA, except with uracil (U) replacing thymine, and is typically single-stranded. The adenine now pairs with uracil whenever the RNA becomes double-stranded. The sugar for RNA differs from the sugar

of DNA and is known as ribose. RNA is the genetic material for most viruses and microorganisms and is found in a variety of forms. Messenger RNA (mRNA) is created from DNA. An enzyme known as RNA polymerase directs the formation of mRNA. RNA polymerase unwinds the DNA, which serves as a template for the new mRNA. The mRNA is a very unstable molecule and typically does not last for any significant amount of time. The purpose of mRNA is to serve as a template for proteins. When mRNA is made from DNA, it contains all of the extraneous material contained in DNA. DNA is composed of two types of sequences of nucleotides: exons and introns. Exons are regions that are genes and code for specific proteins. Introns are regions of DNA that are not made into RNA and do not code for specific proteins. The introns must be spliced (excised) from the mRNA before the mRNA goes to the rRNA in order to create proteins. Transfer RNA (tRNA) are free-floating RNA molecules which carry amino acids that are later attached together to form proteins. Ribosomal RNA (rRNA) are attached to structures within the cell called ribosomes. rRNA "reads" the mRNA and utilizes the mRNA as a template to create proteins, strings of amino acids. Proteins that perform a function on other molecules are known as enzymes. This process of going from DNA to RNA to protein is known as the "Central Dogma."

PCR imitates *in vitro* the method of replication for DNA in nature. In nature, DNA replication is semi-conservative. Semi-conservative means that from one double-stranded DNA, two double-stranded DNA molecules are created where each possesses one newly created strand and one old strand from the original DNA molecule (15). The molecule is separated by an enzyme, known as DNA gyrase, to form an oval, single-stranded "bubble." The bubble has single -stranded DNA on the top and bottom, and to

either side is double-stranded DNA. In the bubble, an enzyme known as DNA polymerase begins to create a new strand of DNA by matching nucleotides to the sense strand and the antisense strand. The new strand of DNA and its corresponding old strand (either the sense or antisense strand) comprise a new DNA helix, resulting in two new helices.

During the replication process, errors in creating DNA, or mutations, can occur. When a single nucleotide in the new strand is altered to a different nucleotide (one that does not match the corresponding nucleotide on the other strand), this is known as a base substitution mutation. These types of mutations can have no effect on the organism, or can have a positive or negative effect. The effect of mutations is not predictable, but can be determined by intensive study of the mutations. Another type of mutation is a frameshift mutation. In this case, extra nucleotides are inserted into the sequence or nucleotides are deleted from the sequence. The number of nucleotides inserted or deleted is not fixed, and can be as few as one nucleotide or even as large as a few hundred nucleotides. This insertion or deletion causes the forthcoming nucleotides to shift positions up or down and alters the codons. When the codons are altered, the sequence of amino acids in the resulting proteins is severely changed. The proteins can end early, extend further, or be composed of different amino acids that is natural. Frameshift mutations typically have negative effects due to the mis-formed proteins.

DNA polymerase may be obtained from many different organisms. As such, the polymerases can have different properties. Some polymerases, such as the *Taq* polymerase from the *Thermus aquaticus* bacterium, are "error-free," meaning they possess a proofreading ability that allows the polymerase to correct misplaced

nucleotides in creating new DNA. *Taq* polymerase typically has a low error rate (1.1 x 10^{-4} base substitutions/bp), dependent upon the availability of nucleotides and the length of the DNA. *Taq* polymerase is also a heat-resistant polymerase, since the bacterium *Thermus aquaticus* is a thermophile (heat-loving bacteria). Heat-resistance is an advantage for this bacterium found in thermal hot springs A DNA polymerase derived from *Escherichia coli* (*E. coli*) denatures in heat and is not very effective. Another popular "error-free" polymerase is the Vent polymerase, from the *Thermococcus litoralis* bacterium with an average error rate of 1.9 x 10^{-5} errors/bp. Polymerases exist that do not possess a proofreading ability, thus have a high error rate and are not used in PCR experiments.

Kary B. Mullis, while working for Cetus Corporation, developed the concept of PCR in 1985 for which he won the Nobel Prize in Chemistry in 1993 (9, 23). Various other scientists at Cetus later developed the methodology (31). In performing PCR, a scientist decides on a sequence of the DNA that is of interest, the "target". The target is typically an unknown sequence of DNA, although as the various genome projects progress the sequences shall be known but not necessarily the function, and the target is needed in large quantities for research. In cases where the nucleotide-by-nucleotide sequence is of interest, such as trying to determine mutations, targets can be inserted into loose pieces of DNA sequences in order to be replicated accurately. Two primers are created to flank the target sequence. Primers are sequences of DNA that are usually 20 nucleotides long. Short primers (e.g. 4 bases long) tend to hybridize (attach) to multiple areas in the genome. If a primer is too long, there is a greater probability that it will circle and hybridize to itself, making it inactive for the PCR. One primer is the

complement to the sequence to the left of the target and the other the complement of the right. In cases where the flanking regions are unknown, known sequences of DNA can be inserted around the target to allow the primers a place to attach. Primers have to be chosen carefully such that the sequence of nucleotides is not a sequence of repeats, e.g. ATATATAT. Repeats occur throughout genomes and are quite frequent. Utilizing a repeat sequence as a primer would incur many extraneous segments of DNA being replicated. The two primers cannot be complements of each other, otherwise the two primers would hybridize to each other and form what are known as primer dimers. Primer dimers do not allow the DNA to elongate since they do not attach to the DNA of interest.

A DNA polymerase is utilized to extend the primers in the direction of the target sequence. Extending the primer involves the polymerase attaching to the end of the primer and continuing with the replication process in the direction of the target. For the primer to attach to the flanking regions of the target sequence, the PCR mixture must be heated to allow the DNA molecules to denature into single-stranded molecules. Once the denaturing is completed, the primers can attach (or anneal) and extension begins. Denaturing typically occurs at 93-94° C, annealing occurs at 50-65° C, and polymerizing, or extending, occurs at 72° C. These temperatures can vary slightly depending on the concentration of the ingredients, the length of the primers, the length of the target, and which polymerase is used. One cycle of PCR is the completion of denaturing, annealing, and extending. After 20-25 cycles, the target sequence has been exponentially replicated and the process is halted. The number of copies of the target in a PCR experiment goes through two phases of growth: exponential and linear. The exponential phase of growth

occurs during the early cycles due to a plentiful supply of enzyme (polymerase), primers, and free nucleotides. The exponential phase transitions into a linear growth as the free nucleotides and primers decrease in number and the enzyme starts to lose its potency. The experiments are usually stopped before going into the linear phase, since the growth in the number of copies of the target is not enough to justify the extra time required.

In PCR experiments, the reaction takes place in an eppendorf micro-test tube. A single PCR machine can typically hold 16-24 tubes. The ingredients in the test tubes determine the efficiency of the PCR. One example of a mixture used for PCR is multiple copies of the genome containing the target in 1.0 μl (microliter) of solution are inserted into the test tube using a micropipette, along with 0.4 μl of the nucleotide mixture, 0.2 μl of the primers, and 0.1 μl of polymerase. The nucleotide mixture contains equal quantities of the four nucleotides, and the primer mixture contains equal amounts of both primers. The final mixture contains a buffer to bring the volume to 10 μl in order for the process to occur. The exact concentrations of the ingredients and the final volume depend upon the results desired and on each individual researcher.

1.2 Prior Methods

Before Kary Mullis developed the concept of PCR, identifying the presence of a specific segment of genetic code was a laborious task and there were no methods available with the ability to quantify the number of copies. To detect if the target sequence was present in a DNA genome (dichotomous response), a Southern blot had to be performed. A Southern blot is a method in which the entire DNA genome is exposed

to restriction endonucleases. Genomic DNA can be obtained from a variety of organic sources, e.g. blood, muscle, skin, etc., and the DNA isolated from the source.

Restriction endonucleases are enzymes that cut (or cleave) the DNA in very specific locations, one enzyme is the endonuclease *Eco*RI, derived from the bacteria *E. coli,* that cleaves the DNA between adjacent A and G nucleotides. The reason the genome is cut is when the molecule is whole, it tends to aggregate and clump together, making it difficult to determine the presence of any sequence of DNA. After the genome has been exposed to multiple endonucleases, the resulting fragments are run out on an electrophoresis gel.

Gel electrophoresis is a method in which the DNA fragments are placed in wells at one end of an agarose gel (a rectangular jelly-like substance), resting in solution with negative electrodes placed on the end of the gel containing the wells and positive electrodes at the other. The negative electric current in the solution causes the DNA to move down the gel away from the negative charge (since DNA has a slight negative charge). The cut DNA moves down the gel and leaves a smear when viewed under ultraviolet light. A dye, placed in one of the wells, is much smaller than any of the DNA fragments, allowing it to move down the gel rapidly. The position of the dye is monitored and the electrophoresis is stopped when the dye reaches the end of the gel so that the DNA does not run off the end of the gel. After the electrophoresis is completed, the DNA is transferred to a special nylon membrane by capillary action. When the DNA is transferred to the membrane, it becomes single-stranded, i.e. the double-stranded DNA separates into two distinct strands. Once the DNA is attached to the membrane, the membrane is placed in a solution containing a radioactive probe that is the complement to

the target. After soaking in solution, the membrane is rinsed so that any unattached probe does not create unnecessary noise. If the probe has attached to the DNA, the radioactivity can be seen under ultraviolet light. If the probe attaches to the DNA, the target sequence is in the genomic DNA. If the excess probe had not been washed off, the entire membrane would "glow" under the ultraviolet light, thus making it impossible to detect the presence of the target.

Another previous method to PCR is the Northern blot method. Northern blots are very similar to Southern blots except Northern blots utilize RNA instead of DNA. Messenger RNA (mRNA) is extracted from cells and placed into a gel with ethidium bromide in it. All cells contain mRNA; therefore any type of cells may be used. After running the gel electrophoresis, the RNA is transferred to a membrane and exposed to a radioactive probe, specific to the desired target. After washing the membrane to get rid of the excess, if the probe can be seen to have attached to the filter under ultraviolet light, the target sequence is said to be in the mRNA.

1.3 Current PCR Methods

There are two modifications to the traditional PCR method that are very popular and used in addition to the traditional method developed by Cetus. One of the modifications is called reverse transcriptase PCR. Reverse transcriptase PCR begins with RNA instead of DNA. DNA polymerase cannot replicate RNA; therefore an additional enzyme is needed to transcribe the RNA into DNA. The additional enzyme is called reverse transcriptase, which uses a strand of mRNA as a template and creates DNA. The newly created DNA is known as complementary DNA (cDNA), since it is a complement

to the RNA. The cDNA formed does not contain the introns that the original genomic DNA does. Once the cDNA has been made, the PCR can continue as in the traditional method. This allows a researcher to detect the presence of viruses, whose genetic material is RNA, such as HIV.

Another modification is the real-time PCR method. Real-time PCR utilizes a probe that is the complement of a section of nucleotides within the target sequence. The probe has a fluorochrome attached to each end, a reporter and a quencher. The reporter fluorochrome emits light of a specific range in the color spectrum and the quencher fluorochrome prevents the reporter from emitting light. As the polymerase is extending the primer to create the new DNA and it comes to the probe, the polymerase releases the nucleotides in the probe. As the nucleotides are released, the two fluorochromes become farther apart and the reporter can begin to emit light. Resting on top of the PCR machine is a high-speed laser camera, which monitors the amount of light being released. Other sequences with known copy number (number of copies of the sequence before PCR), called standards, are run simultaneously in adjacent wells with the target. Based upon when the light starts to become significantly different from the initial light emission, a researcher can determine how many copies of the target were present before PCR (6). Real-time PCR is commonly combined with reverse transcriptase in order to detect the presence of viruses and diseases and estimate the level of infection.

Real-time PCR was developed in order to have an accurate way of estimating the initial number of copies of the target is present. In order to estimate the initial copy number, an understanding of the changes occurring cycle-by-cycle is needed. As the reaction begins, the copy number, C_T, grows slowly for the first 1-5 cycles, then explodes

with the exponential phase. After approximately 25 cycles, the reaction enters the linear phase of growth. By allowing a computer to monitor the reaction through a high-speed camera, the computer can halt the reaction before the experiment reaches the linear phase of growth.

1.4 Data Collection

There are essentially two methods to collecting data in PCR experiments: end-point assay and cycle-by-cycle assay. End-point assays are derived from the traditional PCR method. Some PCR experiments are only trying to determine if a certain sequence of nucleotides is present in the genome (yielding a dichotomous response). Other experiments attempt to quantify the amount of initial copies of the target present. In either case, presence or quantity is determined after the PCR is completed.

In determining the presence of the target, the PCR product (the amplified target sequence) is run on a gel with gel electrophoresis. Upon completion of the electrophoresis, the molecules are transferred onto a thin nylon membrane using an osmosis technique. Once the molecules are on the membrane, an antibody, specific for the target sequence, is exposed to the membrane for approximately 2 hours, and the excess antibody is washed off. The membrane is then stained with ethidium bromide, which stains the antibody, and can be detected using ultraviolet light. If a light band is detected under ultraviolet light, then the target is said to be present in the genome.

In determining the number of copies of the target present, the free nucleotides utilized are radioactively labeled. Thus the new DNA molecules contain radioactive bases, which can be detected using a spectrometer. Again, the PCR product is run on a

gel, and the band containing the products is identified. The band is identified because that band is larger in width than any other band on the gel, since presumably the target was exponentially amplified compared to any miscellaneous DNA. Once the band is identified, the band is cut out of the gel, and dissolved into liquid form. A test tube containing the liquid form is placed into the spectrometer, which then gives the concentration of the liquid. The researcher can then apply a formula to determine approximately the number of copies of the target are present.

In the cycle-by-cycle method of collecting PCR data (real-time PCR), data is collected during each cycle of the reaction using high-speed laser cameras and computers. A camera is attached on top of the PCR machine to monitor the reaction. Each reaction is contained within a well, with each well being contained in a plate. A plate is a rectangular apparatus containing 96 indentions called wells, with each well containing a PCR experiment. This allows the researcher to perform an increased number of experiments simultaneously. The camera records each cycle to the computer, which displays the data as a change in the reaction based upon a baseline established during the first 5 cycles of PCR. Although 96-well plates were initially developed for the cycle-by-cycle collection of data, end-point PCR machines have been developed which accept the plates.

1.5 Types of Data

There are different types of data that are collected from PCR experiments. In traditional PCR, the data is rather subjective. The researcher typically does not know the initial number of copies of the target, but only desires to verify the presence of the target.

Therefore, if there is product that has been amplified at the end of the experiment, the target was present. However, if the free-floating nucleotides contain radioactive phosphorus, the amount of final product can be quantified using a spectrophotometer.

Traditional PCR utilizing the radioactive phosphorus and simultaneously running standards will give an estimate of the initial number of copies of the target. The resulting data consists of the initial copy numbers of the target, the final copy numbers of the standards and the target, and any defining points of interest, such as tube number. The tube number might be of interest to see if there is any between-tube variation.

In real time PCR, the primers are modified in order to emit light during the experiment. A probe is utilized which hybridizes (attaches) to the center of the target and contains fluorochromes at either end, one being an emitter and the other a repressor. The two fluorochromes cancel each other out when they are at either end of the probe so that no light is actually emitted. As polymerase extends the primer and reaches the probe, it begins to eat away at the probe nucleotide by nucleotide so that the polymerase can add nucleotides to the primer to create the new DNA molecule. As the fluorochromes are released from the probe, light begins to be emitted and can be measured. A high-speed camera measures the amount of light emitted at each cycle, and the data actually reported is the change in fluorescence, ΔR_n. A baseline measurement is taken as an average of the fluorescence detected in the first 5 to 10 cycles and the change is determined to be difference from the baseline. Then a formula can be applied to determine the copy number of the target during each cycle.

Reverse transcriptase PCR has the same types of data as traditional PCR. When real-time PCR is combined with reverse transcriptase, or even real-time PCR alone,

additional data is available. Standards of known copy number are run at the same time as the target, whose initial copy number is unknown. This allows a comparison of the target to the standards to estimate the initial copy number of the target. A computer generates the graphs of the change in fluorescence of each standard and the target, and the researcher can observe which standard line is most like the target's line.

1.6 Applications of PCR

PCR is utilized in many industries and fields for a variety of reasons. One reason is to detect the presence or quantify any particular virus or other genetic material. However, in small samples, such as a blood sample drawn from a patient, the amount of virus present in the sample is not detectable by current methods. In order for a virus to be detected, there must be at least 10^{13} to 10^{14} molecules present in the sample. In the course of normal illnesses and diseases, the virus is present in much lower quantities. PCR can amplify a portion of the virus genome to detect its presence. PCR does not have to amplify from the entire genome. In some bacterium, a plasmid (extraneous circle of DNA) may be present carrying genes obtained from other bacteria and sources, thus conflicting what to amplify. Yet a section of DNA from the flagellum (tail) can be amplified to detect the presence of a specific bacterium.

The method is commonly utilized to detect the presence or absence of genetic anomalies, diseases, etc. If the sequence of a genetic disorder, such as breast cancer in humans or hip dysplasia in dogs, is known, PCR can be used to determine if a person is a carrier for the disease. The genomic DNA can be isolated as exposed to the PCR process in which the primers are those that would normally flank the target gene. If a product is

amplified, it would be safe to say that the gene of interest in present in the genome. Gene counseling is based upon this idea of knowing whether a person is a carrier for a disease or not. A person is a carrier for a genetic disorder if they contain the gene in their genome, but do not show the symptoms of the disease. PCR can be used to determine the virus load of an HIV-infected person by extracting the RNA from a blood sample and amplifying it. If there is a sequence specific to the anthrax virus, detecting if there is any virus present would encompass only a few hours, compared to a few weeks with current methodology. There are many uses for PCR, once the initial obstacle of sequencing DNA and RNA is overcome.

The genetic material of viruses is typically RNA, and PCR is modified to allow the reproduction of a section of RNA. This modification requires an enzyme called a reverse transcriptase that converts the RNA back into DNA, and polymerase can begin reproducing the DNA. PCR is one way to detect the presence of antibiotic-resistance in bacteria by attempting to amplify the resistant section of DNA. If the bacteria are antibiotic-resistant, then the fragment will replicate and there will be hundreds of thousands of copies of the section at the end of 25 PCR cycles.

Lewin *et. al.* (10) used real-time PCR in combination with reverse transcriptase to detect HIV presence in antiretroviral treatment patients. The Human Immunodeficiency Virus can exist in small "pockets" after effective treatment, allowing regrowth of the virus. Detection of these pockets has proved difficult, yet mRNA from the virus exists in the bloodstream of the patients. A real-time reverse transcriptase PCR was used to detect minute levels of the HIV mRNA, and discovered that patients with latent areas of live HIV can be detected with the PCR method when no positive plasma HIV exists.

Machida *et al*. (12) used real-time PCR in the detection of cytomegalovirus (CMV), which is a fatal viral complication of a bone marrow transplant, and found that real-time PCR was a more sensitive and rapid method than conventional methods in determining susceptibility to CMV. The amount of CMV was able to be quantified using real-time PCR and identified patients at high risk for complications. For further examples of PCR utilized in medical fields, see 2 – 4, 7, 8, 14 – 18, 20, 22, 25 – 27, 29, 30, 32 and 33.

Chiang *et. al.* (1) proposed using PCR to map gene deletions in the *Drosophila* genome, which means that when a deletion occurs in a genome, it can be mapped, or identified, with a certain area of the chromosome. Liu *et. al.* (11) utilized PCR to differentiate between heterozygotes and homozygotes. A gene is present in many forms called alleles and a genome typically has two copies of the gene. If the two copies of the gene are the same (same alleles), then the gene is said to be homozygous. If the alleles are different, then the gene is said to be heterozygous. The differences between alleles are a section of the DNA composed of different nucleotides, or the nucleotides are in a different order. See also 16 and 21.

1.7 Goals

In this thesis, we will understand a mathematical model for PCR and perform simulations of PCR to develop an estimate of the initial copy number of the target.

CHAPTER 2

BRANCHING PROCESSES

2.1 Branching Processes

The stochastic model describing the PCR mechanism is branching processes. Branching processes are methods to represent sizes of populations where the individuals in the population reproduce a random number of offspring at the end of the lifetime, independent of each other and of the number in the population. The number of offspring follows a certain probability distribution, and the total population size after each time point is a branching process. An integral part of branching processes is that the replication of each individual in the population is independent of any other individual in the population and also independent of previous replications of the ancestors.

Biologically, many systems follow a branching process model. Population dynamics can follow a branching process. Amoeba populations follow a branching process under certain conditions when each amoeba splits into "daughter" amoebas. At the initial time point, time 0, there are a fixed number of ancestors in the population, represented by Z_0. Each individual in the population survives for one unit of time (being generations, cycles, etc.) and at the end of that unit of time reproduces a random number of offspring. The number of offspring will take on values that are non-negative integers. In many biological systems, the maximum number of offspring is limited by a value less than infinity. There is a probability distribution associated with the number of children born for any one individual in the population. The simple case for

branching processes is where there is one ancestor in generation 0, and the complex case where there is more than one ancestor in generation 0.

2.1.1 Single Ancestor Branching Processes

When there is only one ancestor in generation 0, the number of individuals in the first generation is the number of children of the initial ancestor, represented by Z_1. The probability that the first generation will have 0 members is

$$P(Z_1 = 0) = p_0$$

and the probability that Z_1 will contain 1 member is

$$P(Z_1 = 1) = p_1.$$

Furthermore, the probability that the first generation will contain i individuals is

$$P(Z_1 = i) = p_i, i = 0,1,\ldots.$$

The second generation, Z_2, contains all of the offspring from individuals in Z_1. The probability of the j^{th} individual in Z_1 having i offspring is still

$$P(\xi_{1,j} = i) = p_i, i = 0,1,\ldots,$$

where $\xi_{1,j}$ is the number of offspring of the j^{th} individual in the first generation. Each individual has the same offspring probability distribution for the entire lifespan of the population. The individuals in the first generation reproduce independent of each other and independent of the number of individuals in Z_0. Therefore, Z_2 is

$$Z_2 = \sum_{j=1}^{Z_1} \xi_{1,j}$$

where $\xi_{1,j}$ is the number of offspring produced by the j^{th} parent in the first generation. In general, the number of individuals in the n^{th} generation is

$$Z_n = \sum_{j=1}^{Z_{n-1}} \xi_{n-1,j} \ .$$

Then the process of $\{Z_n: n \geq 0\}$ is called the branching process for the system being modeled. In the event that $Z_n = 0$, the population is said to be extinct, and implies that $Z_{n+1} = 0$ also.

Each generation of the population has a certain mean dependent upon the offspring probability distribution. The expected value for the first generation Z_1 is

$$E(Z_1) = \sum_{j \geq 0} j \cdot p_j = m \ ,$$

which will be used in the means of further generations, such as Z_2. The mean for Z_2 is as follows:

$$E(Z_2) = E(\sum_{j=1}^{Z_1} \xi_{1,j}) = m^2 \ .$$

By induction, the mean of Z_n is

$$E(Z_n \mid Z_0 = 1) = m^n \ .$$

The expected value of Z_n being m^n indicates a geometric growth of the population.

The mean of the n^{th} generation, m^n, infers the type of branching process that is being modeled. If $m < 1$, then as n goes to infinity, m^n goes to zero and the population will become extinct. In this case, the branching process is called a subcritical branching process. If $m = 1$, then as n goes to infinity, m^n goes to 1 and the population still will eventually become extinct if $p_1 \neq 1$. For $m = 1$, it is assumed that $p_1 \neq 1$. In this scenario, the branching process is known as a critical branching process. If $m > 1$, then as n goes to infinity, m^n goes to infinity and the population explodes with some probability known as *(1 – extinction probability)*, where the extinction probability is the probability that the

population will become extinct. This branching process is known as a supercritical branching process.

In a subcritical branching process, the population will become extinct with probability 1. For a critical branching process, the population will become extinct, assuming $p_1 \neq 1$, with probability of 1. The supercritical branching process population can either become extinct or can continue growing. For the supercritical branching process, an additional parameter can be calculated, the extinction probability. This is the only branching process for which an extinction probability is logical, since the probability for extinction of both the subcritical and critical is 1.

The extinction probability is not dependent upon the number of individuals in the initial generation Z_0. The extinction probability is found from the probability generating function of the process. For the first generation when $Z_0 = 1$,

$$f_1(s) = E(s^{Z_1}), \quad 0 \leq s \leq 1$$

$$= \sum_{j \geq 0} s^j p_j .$$

For the second generation,

$$f_2(s) = E(s^{Z_2})$$

$$= E\left[E\left(s^{\sum_{j=1}^{Z_1} \xi_{1,j}} \mid Z_1 \right) \right]$$

$$= f(f(s)).$$

By induction, the generating function for the n^{th} generation is

$$f_n(s) = f(f_{n-1}(s)), n \geq 1.$$

The extinction probability, q, is the probability that the population will die out:

$$q = P(Z_n = 0 : n \geq 1),$$

where q satisfies the equation

$$q = f(q).$$

2.1.2 Multiple Ancestors Branching Process

Many branching processes do not begin with only one individual in the initial generation. Bacterial colonies start with many bacteria and growth to hundreds of millions of cells. As when there is only a single individual in the initial generation, the probability that the j^{th} individual in Z_0 will have i offspring is

$$P(\xi_{0,j} = i) = p_i, i = 0,1,\ldots$$

and this distribution is the same for following generations in the population. Each individual produces offspring independently of the other members of that generation and independent of previous generations. The number of individuals in the first generation is given by:

$$Z_1 = \sum_{j=1}^{Z_0} \xi_{0,j}, where\, Z_0 > 1,$$

where $\xi_{0,j}$ is the number of offspring from the j^{th} individual in the initial generation.

Each generation in a multiple ancestor branching process has a modified mean when compared to the means of the generations of a single ancestor branching process. The expected value for the first generation Z_1 is

$$E(Z_1) = E\left(\sum_{j=1}^{Z_0} \xi_{0,j}\right) = Z_0 m$$

The mean number of individuals in the second generation is

$$E(Z_2) = E(\sum_{j=1}^{Z_1} \xi_{1,j})$$

$$= Z_0 m^2 .$$

By induction, the mean number of individuals in the n^{th} generation is

$$E(Z_n) = Z_0 m^n .$$

The extinction probability for the multiple ancestor branching process using independence of the lines of descent can be calculated as follows:

$$P(Z_n = 0 \text{ for some } n \geq 1 \mid Z_0 = z_0) = q^{z_0},$$

where q is the solution for $f(q) = q$.

In the next section, since q is an important parameter in the supercritical branching process, we will investigate the estimation of q.

2.2 Estimation of Extinction Probabilities

In estimating the extinction probability of a supercritical branching process, a simulation needs to be developed which utilizes the properties of the process. As stated earlier, the growth of a branching process is geometric, which has the following distribution:

$$P(N = n) = \theta^n (1 - \theta), n \geq 0, 0 \leq \theta \leq 1,$$

where n represents the number of offspring an individual will have before failing to have offspring and θ represents the probability that an individual will have children. With the

number of offspring represented by a geometric distribution, the mean m is represented in terms of θ:

$$E(N) = \sum n\theta^n (1-\theta)$$

$$= \theta(1-\theta)\sum_{n\geq 0} n\theta^{n-1} = \theta(1-\theta)\sum_{n\geq 0}\frac{d}{d\theta}\theta^j$$

$$= \theta(1-\theta)\frac{d}{d\theta}\left(\sum_{n\geq 0}\theta^n\right) = \theta(1-\theta)\left(\frac{d}{d\theta}\left(\frac{1}{1\theta}\right)\right)$$

$$= \theta(1-\theta)\left(\frac{1}{1-\theta}\right)^2 = \frac{\theta}{1-\theta}.$$

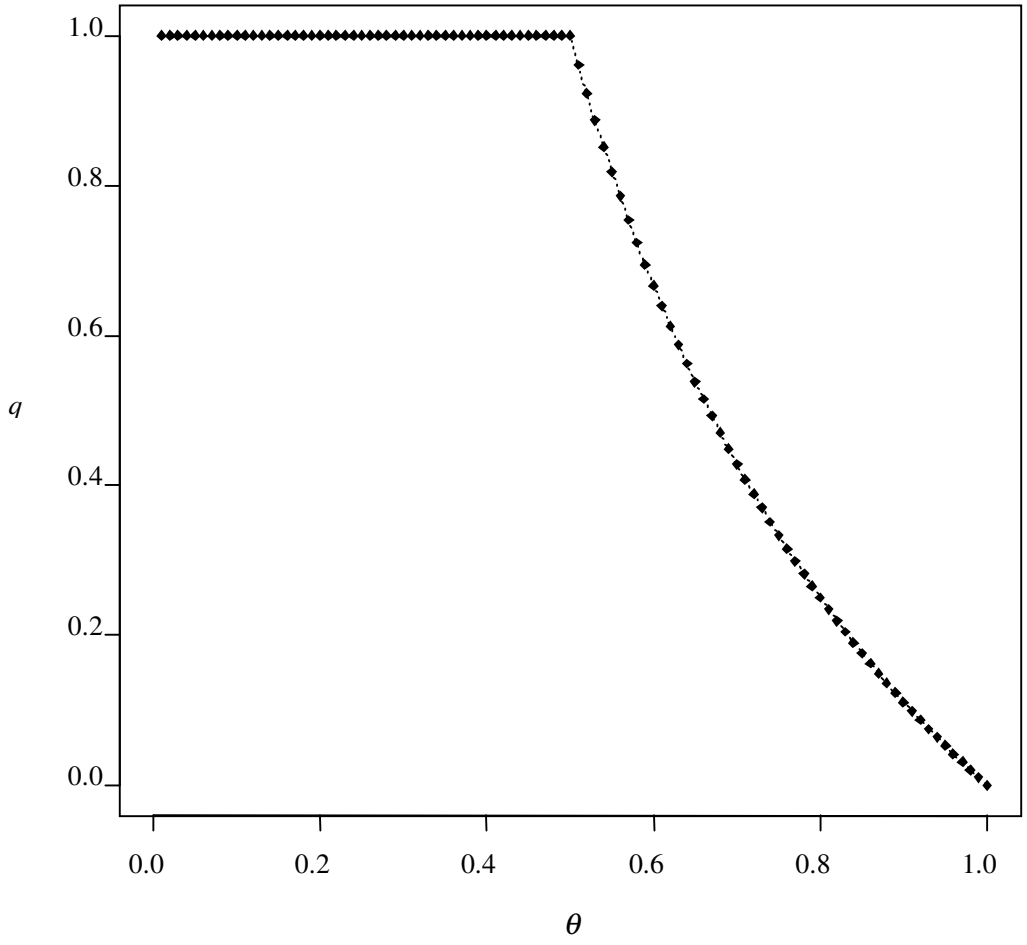

Figure 2.1: Probability of Reproducing versus Extinction Probability

In order to represent a subcritical branching process with $m < 1$, θ needs to be less than 0.5 and the extinction probability is 1. For a critical branching process with $m = 1$, θ must equal 0.5 and the extinction probability is 1. For a supercritical branching process with $m > 1$, θ must be greater than 0.5 and the extinction probability is less than 1. This phenomenon is demonstrated in Figure 2.1. To obtain an extinction probability less than 1, a supercritical branching process must be simulated. In order to compare the estimated extinction probability to the actual extinction probability for a given θ, the actual extinction probability q was determined solving the equation:

$$q = f(q) = \frac{1-\theta}{1-q\theta}.$$

This leads q to be the solution of:

$$\theta q^2 - q + (1-\theta) = 0,$$

which is the extinction probability for any number of generations.

As an example of a simulation to estimate the extinction probability, the initial generation Z_0 was fixed at 2 and θ fixed at 0.6. The program used for the simulation is shown in Appendix B. For each individual in Z_0, a uniform(0,1) random variable was picked and used to obtain a geometric random variable. The transformation used for the geometric required that the parameter sent to it be $(1-\theta)$, thus 0.4 was used in the simulation. The actual extinction probability q for $\theta = 0.6$ is $q = 0.\overline{66}$. The results shown in Figure 2.2 indicate that as the number of generations increases, the estimate of the extinction probability becomes closer to the actual extinction probability.

2.3 PCR and Branching Processes

Theoretically, PCR in the exponential phase of growth should follow a modified branching process model. As the process proceeds, the model for PCR changes due to limiting factors, such as nucleotides, enzyme, and primers. A PCR experiment begins with between 10 copies (low-copy number PCR) to thousands of copies of the target sequence. In a branching process, the number of offspring an individual can produce 0 offspring is between 0 and 1.

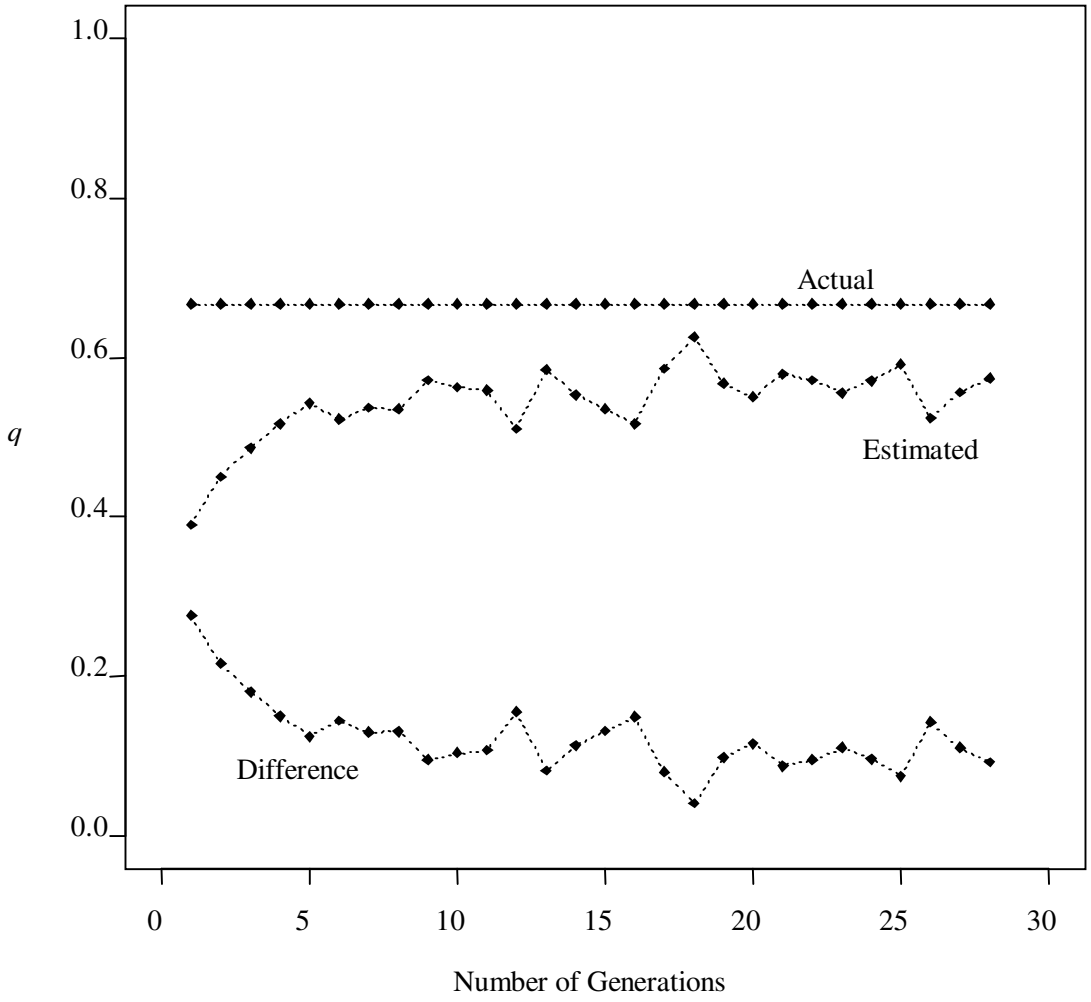

Figure 2.2: Estimated Extinction Probability from Simulations

In PCR, the probability that an individual will produce zero offspring is 0, since the strand of DNA does not "die out" at the end of a cycle. The number of new DNA strands that a strand can produce is either one or two strands: one if the strand does not replicate and two if the strand does replicate.

Let p_1 be the probability that the DNA replicates, then $1 - p_1$ is the probability that the DNA does not replicate. The mean for the first generation is

$$E(Z_1 \mid Z_0 = 1) = 2p_1 + 1(1 - p_1)$$

$$= 1 + p_1 > 1.$$

In theory, p_1 may be 0, leading the expected value to be 1, indicating a critical branching process. If p_1 were 0, then a molecule of DNA will always produce 2 molecules, indicating a perfect reproduction always occurring. However, PCR is a method of mimicking DNA replication utilizing a mechanism, the polymerase, that does not always perfectly replicates the DNA. The primers do not always attach to the DNA to allow the polymerase to work. Therefore p_1 may be a small number, but never 0 and the expected value of the mean of the first generation is greater than 1, the branching process that represents PCR is a supercritical process, and p_1 is called the efficiency of the PCR. The efficiency of PCR is affected by the polymerase that is utilized. There are some polymerases that exist that attach to the template and replicate more frequently and efficiently that others.

PCR follows the branching process with

$$m = 1 + p_1$$

during the exponential phase of replication. However, there are limiting factors involved in the experiment, such as the amount of free nucleotides and the amount of polymerase.

These limiting factors cause the PCR branching process to change into a linear phase of growth with a very small slope, eventually leveling to an asymptote. PCR experiments are halted before the linear phase begins. Therefore, we are concerned with the exponential phase of growth.

CHAPTER 3

MODELING AND SIMULATION

3.1 Modeling

In this chapter, we develop a statistical model for estimating the initial number of copies in a PCR experiment when the data arise from an end-point assay. This model was developed by Vidyashankar (28). We first state the assumptions that lead to the statistical model.

Assumptions:

A1. The standard and the target amplify at the same rate.

A2. All molecules in the target and in the standard amplify at the same rate during every cycle.

For assumption A1, biologically polymerase operates at the same rate for any sequence of DNA that it is replicating, providing there is an ample supply of free nucleotides available. From this biological premise, the standard and the target will amplify at the same rate because the same polymerase is being utilized. For assumption A2, the molecules for the target and the standard are independent when there is ample supply of polymerase, free nucleotides, and primers. During the exponential phase of growth for PCR, there are amply quantities of these ingredients and the molecules do not have to compete for them.

Let $X_n(j),\ 1 \leq j \leq K$, denote the number of standard molecules from the j^{th} well at the end of K cycles arising from $X_0(j)$ number of initial molecules. Let $Z_n(j),\ 1 \leq j \leq K$,

denote the number of target molecules in the j^{th} well at the end of the K^{th} cycle arising from $Z_0(j)$ initial molecules. Note that $Z_0(j) = Z_0$ for all j. From Chapter 2 we know that $E(X_n(j)) = (1 + p_1)^n X_0(j)$ where p_1 is the probability that a DNA molecule will not split during a given cycle. By assumption A1, $E(Z_n(j)) = (1 + p_1)^n Z_0(j)$. Empirically it has been observed that a plot of $X_0(j)$ against $log(Z_n(j) / X_n(j))$ is linear (see Vidyashankar (28)). Motivated by this reason, we use the model

$$ln\left(\frac{Z_n}{X_n}\right)(j) = \beta_0 + \beta_1 ln(X_0)(j) \qquad 1$$

where β_0 and β_1 are the regression parameters. Now, extending the regression line to zero, gives the value of X_0 for which $Z_n = X_n$. If p_1 is small, then one can estimate Z_0 to be the value of X_0 that will make the LHS of (1) to be 0. From this, it follows that the estimate of the initial copy number for the target is

$$\hat{Z}_0 = \exp\left(\frac{\hat{\beta}_0}{-\hat{\beta}_1}\right). \qquad 2$$

The standard error of $\hat{Z}_0$ is harder to compute, however we provide some approximations using standard large sample theory. We note that these numbers are smaller than the 'true' standard errors.

3.2 Simulations

The data were generated using Visual C++ 6.0, and then analyzed using SAS version 8.0. The initial set of data was generated using the branching process code shown in Appendix C. The SAS program used to analyze the data is shown in Appendix D. A variety of methods to generate the data were examined, beginning with idealized results and leading to results more representative of actual experiments.

3.2.1 Fixed probability model

Researchers employ polymerases that have a high probability of reproducing, yet the probability is also dependent on the composition of nucleotides in the sequence. Therefore an attempt is made to have targets that will have a probability of reproduction similar, if not equal, to the probability for the target. As an initial test of the model, data were generated where the probability of reproducing for both the target and all standards was 0.85. The initial copy number was set at 50 for the target, with the standards being 30, 40, 50, 60, and 70. In running the simulations, there were 6 plates with 5 wells each, resulting in 30 lines of data for each of the 10,000 simulations. Each simulation was analyzed using PROC MIXED (ran 10,000 times), the estimates for each simulation were outputted, and the means of the resulting estimates over the 10,000 simulations were determined using PROC MEANS. The model from equation 1 was utilized for PROC MIXED with the plate and well being random effects. This model and random effects will be utilized for all future analyses.

The results of the PROC MEANS for 10 generations of PCR are shown in Table 3.1.

Variable	Mean	Standard Deviation	Minimum	Maximum
$\hat{\beta}_0$	3.68233056	0.23160882	2.86321929	4.66173437
$\hat{\beta}_1$	-0.93155109	0.05903379	-1.17126046	-0.71691392
$ln(\hat{Z}_0)$	3.95306714	0.01865157	3.87890873	4.02866535
$\hat{Z}_0$	52.10396789	0.97229682	48.37140002	56.18587290

Table 3.1: 10,000 Simulations of Fixed Probability $p = 0.85$ with $n = 10$, $Z_0 = 50$

The mean estimate for the initial copy number of the target was based upon using equation 2 to obtain an estimate of the initial copy number for each simulation, rather than using equation 2 on the overall mean of the logarithmic value. The mean covariances of the plate effect and the well effect over the simulations are 0.00047804 with standard deviation 0.00082633 and 0.0018310 with standard deviation 0.00198526, respectively. The covariances indicate that the plate and well variation do not truly effect the data.

The mean results of PROC MEANS with 15 cycles are shown in Table 3.2.

Variable	Mean	Standard Deviation	Minimum	Maximum
$\hat{\beta}_0$	3.56000385	0.23235960	2.73881543	4.52194494
$\hat{\beta}_1$	-0.89552318	0.05919567	-1.14841929	-0.67964871
$ln(\hat{Z}_0)$	3.97560095	0.01949250	3.90267868	4.05057850
$\hat{Z}_0$	53.29225606	1.03989495	49.53495963	57.43067124

Table 3.2: 10,000 Simulations of Fixed Probability $p = 0.85$ with $n = 15$, $Z_0 = 50$

The estimates from the 10-cycle-simulations are closer to the true value of the initial copy number than are the estimates from the 15-cycle-simulations. This indicates that the experiment can be halted 10 – 15 cycles earlier than is traditional (20-25), provided that the probabilities of reproducing are the same from the first cycle. However, in actual experiments, the probabilities are not equal and it takes few cycles to standardize before the exponential phase due to measurement errors in the initial cycles. Hence, 20-25 cycles are used for obtaining data and our simulations justify the experimental evidence.

3.2.2 Normal probability model

The probability that a DNA molecule will reproduce is not constant, but is dependent upon several factors. These factors in PCR are the quantities of the various ingredients, the length of the sequence being replicated, and the composition, or number of each nucleotide, of the sequence. The most common form of error that results in the context of PCR is the pipetting error. To account for these in the simulations, the probabilities were chosen randomly from a distribution, with a different p for each standard and a different p for each target in a well. The derivation of p is as follows:

$$ln\left(\frac{p}{1-p}\right) \sim N(0,\sigma^2)$$

Then, choose X to be a randomly normal variable.

$$X \sim N(0,\sigma^2)$$

$$X = ln\left(\frac{p}{1-p}\right)$$

$$p = \frac{e^X}{1+e^X}.$$

Each simulation was performed 10,000 times with 30 lines of data each (6 plates with 5 wells) with number of cycles being 5, 6, 8, and 10 to see if the model is accurate for small cycle numbers. Each number of cycles was run using X as a random normal variable with mean 0 and standard deviation of one of the following: 2, $\sqrt{2}$, 3, and $\sqrt{3}$. This resulted in 16 datasets of 10,000 simulations. Again, results were analyzed using PROC MIXED, then PROC MEANS to determine the confidence intervals and overall

mean estimates. The results from one of the simulations, $n = 10, \sigma = \sqrt{2}$, are shown in Table 3.3.

Variable	Mean	Standard Deviation	Minimum	Maximum
$\hat{\beta}_0$	3.86194357	10.62718427	-27.00301336	38.01144307
$\hat{\beta}_1$	-0.98074662	2.70211008	-9.60396474	6.74989316
$ln(\hat{Z}_0)$	16.62587047	1334.6936587	-1977.571341	133423.69441
$\hat{Z}_0$	2.870963×10^{159}	.	0.00000000	2.870101×10^{163}

Table 3.3: 10,000 Simulations of Normal Probability, $N(0, \sqrt{2})$, with $n = 15$ and $Z_0 = 50$

The mean covariance parameters for plate and well were 0.00071977 with standard deviation of 0.00131973 and 3.21085337 with standard deviation of 2.11340752, respectively. All 16 sets of data had similar trends.

As can be seen by the mean estimate of the logarithmic initial copy number of the target, the standard deviation is excessive to order to accurately estimate the copy number. The reason for the excessively large standard deviation is when a random normal is chosen with mean 0, approximately half of the random variables will be below zero, and half above. This allows the probability to have a wide range, which may not allow the model to accurately represent the data. Therefore the next method involves randomly selected *p* from a range of possible values.

3.2.3 Wide Uniform Probability Ranges Model

Since the normal probability model ended up being a very poor method of introducing variability into the branching process, it was decided to attempt to introduce variability by uniformly selecting a probability from a certain range. In Mathieu-Daudé

et. al. (13), a researcher can expect at best that a single DNA molecule will reproduce 1.9 new molecules, indicating that the probability of reproduction is at best 0.9. Therefore the ranges for probability were set between 0.9 as a maximum and 0.6 as a minimum. The minimum value was set at 0.6 because researchers desire and strive for high probabilities of reproduction. The number of generations utilized was 8 to determine whether 8 cycles is ample information to predict the initial copy number of the target.

Again, PROC MIXED and PROC MEANS were used to analyze the resulting 5 datasets, with 300,000 lines of data each. The results for the estimate of the initial copy number for the target are shown in tables 3.4 and 3.5.

Range	Mean Estimate, $\hat{\beta}_0$	Standard Deviation, $\hat{\beta}_0$	Mean Estimate, $\hat{\beta}_1$	Standard Deviation, $\hat{\beta}_1$
0.6 – 0.9	3.96518373	3.23341915	-1.01444278	0.83265619
0.7 – 0.85	3.87807007	1.60858375	-0.99177003	0.41433266
0.7 – 0.9	3.91342611	2.10465997	-1.00132941	0.54181529
0.8 – 0.9	3.92962788	1.03204010	-1.00435265	0.26603226
0.85 – 0.9	3.91397777	0.51813809	-1.00040646	0.13345320

Table 3.4: Parameter Estimates for 10,000 Simulations, with $n = 8$ and $Z_0 = 50$

Range	Mean Estimate, $ln(\hat{Z}_0)$	Standard Deviation, $ln(\hat{Z}_0)$	Mean Estimate, $\hat{Z}_0$	Standard Deviation, $\hat{Z}_0$
0.6 – 0.9	3.95562922	18.09427341	2.890729×10^{156}	.
0.7 – 0.85	3.92049517	1.97768767	2.3157348×10^{45}	2.3157348×10^{47}
0.7 – 0.9	3.99594460	11.68558058	5.297483×10^{165}	.
0.8 – 0.9	3.91672892	0.09374277	50.46438164	5.06685973
0.85 – 0.9	3.91315904	0.04174034	50.10051696	2.09664003

Table 3.5: Copy Number Estimates for 10,000 Simulations, with $n = 8$ and $Z_0 – 50$

For the sets of probabilities where the range was small, the estimates of the initial copy number of the target are close to the real value. When the range is wide, such as *0.6 – 0.9*, the estimates for the initial copy number either cannot be estimated using PROC MIXED, or are values vastly outside the acceptable range. The estimates of the logarithmic values when the range is wide are decent estimates, however the standard deviations are so large that the estimates are not practical.

Range	Mean, Plate	Standard Deviation, Plate	Mean, Well	Standard Deviation, Well
0.6 – 0.9	0.00032905	0.00058183	0.31776797	0.23219759
0.7 – 0.85	0.00028762	0.00051314	0.07703285	0.05688375
0.7 – 0.9	0.00025416	0.00045104	0.13349995	0.09871132
0.8 – 0.9	0.00018660	0.00032365	0.03103435	0.02323986
0.85 – 0.9	0.00014759	0.00025794	0.00767181	0.0060149

Table 3.6: Covariance Parameters for Plate and Well Effects, for $n = 8$ and $Z_0 = 50$

The plate covariance parameters shown in table 3.6 are consistent with the plate covariance parameters from the fixed probability model, shown in section 3.2.1, indicating that plate does not have much effect on the outcome of the experiment. However, since there is variability introduced into the probability of reproducing, the well covariance parameter is important in that the values for the model parameters are small and easily altered due to the well covariance. In light of the width of the probability ranges, the next model confines the probability ranges to a maximum width of 0.10.

3.2.4 Narrow Uniform Probability Ranges Model

For the final method of introducing variability in the probabilities, the maximum range of p was set to 0.10, with the minimum 0.80 and the maximum at 0.90. There were

10 ranges of width 0.01, 9 of width 0.02, 8 of 0.03, 7 of 0.04, 6 of 0.05, 5 of 0.06, 4 of 0.07, 3 of 0.08, 2 of 0.09 and 1 of 0.10. Each probability interval was simulated 1,000 times for two cycle numbers, 8 and 10, and for two target initial copy numbers, 50 and 100, resulting in 220 datasets of 30,000 lines of data each. PROC MIXED and PROC MEANS were performed on each of the datasets (MIXED ran 220,000 times and MEANS ran 220 times), and tables 3.7 and 3.8 show examples from each of the 10 ranges of probability for $n = 8$ and $Z_0 = 50$. All sets of data within a certain width of probability had similar results for both $n = 8$ and $n = 10$.

Range	Mean Estimate, $\hat{\beta}_0$	Standard Deviation, $\hat{\beta}_0$	Mean Estimate, $\hat{\beta}_1$	Standard Deviation, $\hat{\beta}_1$
0.80 – 0.81	3.92052880	0.19112118	-1.00216233	0.04871722
0.80 – 0.82	3.90950342	0.25920681	-0.99972725	0.06643497
0.80 – 0.83	3.89992607	0.34749328	-0.99713043	0.08896855
0.80 – 0.84	3.93030718	0.42577839	-1.00399064	0.10892289
0.80 – 0.85	3.88924217	0.52779048	-0.99523429	0.13547944
0.80 – 0.86	3.93340827	0.62355692	-1.00502228	0.15937270
0.80 – 0.87	3.92946161	0.74143272	-1.00454420	0.18953510
0.80 – 0.88	3.93837605	0.84090814	-1.00782025	0.21659108
0.80 – 0.89	3.95282314	0.92828843	-1.00945402	0.23901109
0.80 – 0.90	3.85672044	1.03103632	-0.98442945	0.26431054

Table 3.7: Parameter estimates for 1,000 Simulations, with $n = 8$ and $Z_0 = 50$

The standard deviations of the parameter estimates indicate that the tighter the range of probabilities, the closer the estimates of the initial copy number for the target will be to the exact value.

Range	Mean Estimate $ln(\hat{Z}_0)$	Standard Deviation $ln(\hat{Z}_0)$	Mean Estimate $\hat{Z}_0$	Standard Deviation $\hat{Z}_0$
0.80 – 0.81	3.91207232	0.01528231	50.00829933	0.76432129
0.80 – 0.82	3.91066663	0.02177220	49.94404863	1.08714304
0.80 – 0.83	3.91129248	0.02885436	49.98427640	1.44323638
0.80 – 0.84	3.91490599	0.03746348	50.17953326	1.88129972
0.80 – 0.85	3.90836291	0.04818969	49.87515900	2.40537827
0.80 – 0.86	3.91425567	0.05680771	50.19255290	2.85126016
0.80 – 0.87	3.91208053	0.06457408	50.10722826	3.24376661
0.80 – 0.88	3.90988683	0.07487219	50.03340073	3.75986316
0.80 – 0.89	3.91888355	0.08681639	50.53506042	4.43198265
0.80 – 0.90	3.92006220	0.10300081	50.67381942	5.32189925

Table 3.8: Copy Number Estimates for 1,000 Simulations, with $n = 8$ and $Z_0 = 50$

Table 3.10 shows the mean covariance estimates for the plate effect are similar to those in section 3.2.1, while the well effect is more noticeable as the interval becomes wider. This shows that the smaller the range, outside influences such as well play a lesser role in the outcome of the PCR experiment.

Range	Mean, Plate	Standard Deviation, Plate	Mean, Well	Standard Deviation, Well
0.80 – 0.81	0.00024989	0.00042756	0.00048427	0.00071400
0.80 – 0.82	0.00022589	0.00040130	0.00128891	0.00142833
0.80 – 0.83	0.00022369	0.00038682	0.00282539	0.00274226
0.80 – 0.84	0.00020693	0.00038407	0.00495099	0.00437981
0.80 – 0.85	0.00022188	0.00036457	0.00741870	0.00614725
0.80 – 0.86	0.00022509	0.00039382	0.01051327	0.00831678
0.80 – 0.87	0.00019863	0.00036872	0.01516181	0.01146657
0.80 – 0.88	0.00017578	0.00032452	0.01983830	0.01549978
0.80 – 0.89	0.00017753	0.00034700	0.02506769	0.01912491
0.80 – 0.90	0.00017175	0.00032767	0.03089179	0.02296520

Table 3.10: Covariance Parameters for Plate and Well Effects, for $n = 8$ and $Z_0 = 50$

The parameter estimates and copy number estimates are shown in tables 3.11 and 3.12 for $n = 8$ and $Z_0 = 100$. Again, 1 example from each of the 10 ranges of probability are shown, and all estimates within a range were similar. Results were similar for $n = 8$ and $n = 10$. The standards chosen to compare the target to were 40, 60, 80, 120, and 160. The actual logarithmic value for $Z_0 = 100$ is 4.605170185988.

Range	Mean Estimate, $\hat{\beta}_0$	Standard Deviation, $\hat{\beta}_0$	Mean Estimate, $\hat{\beta}_1$	Standard Deviation, $\hat{\beta}_1$
0.80 – 0.81	4.61135231	0.13035100	-1.00134253	0.02925073
0.80 – 0.82	4.60826464	0.18221910	-1.00067793	0.04094727
0.80 – 0.83	4.61272327	0.25550722	-1.00155817	0.05783624
0.80 – 0.84	4.63514853	0.31971842	-1.00709658	0.07195700
0.80 – 0.85	4.61138339	0.39569635	-1.00092010	0.08916855
0.80 – 0.86	4.64654458	0.47084918	-1.00922179	0.10582502
0.80 – 0.87	4.63849472	0.53935578	-1.00738178	0.12138526
0.80 – 0.88	4.58167156	0.62867451	-0.99533998	0.14172552
0.80 – 0.89	4.58706619	0.67946312	-0.99580220	0.15310734
0.80 – 0.90	4.61635219	0.77095399	-1.00234301	0.17288990

Table 3.11: Parameter Estimates for 1,000 Simulations, with $n = 8$ and $Z_0 = 100$

The estimates indicate that the model gives accurate estimates of the initial copy number of the target, even though the actual value of $Z_0 = 100$ was not included in the values of the standards. The covariance effects for plate in table 3.13 are similar as previous methods of determining the probabilities. The well effects in table 3.13 follow the same trend for $Z_0 = 100$ as in table 3.10 for $Z_0 = 50$.

Range	Mean Estimate $ln(\hat{Z}_0)$	Standard Deviation $ln(\hat{Z}_0)$	Mean Estimate $\hat{Z}_0$	Standard Deviation $\hat{Z}_0$
0.80 – 0.81	4.60531428	0.01341490	100.02339955	1.34159789
0.80 – 0.82	4.60544779	0.02127244	100.05037973	2.12899553
0.80 – 0.83	4.60626022	0.03032251	100.15512441	3.04489612
0.80 – 0.84	4.60346729	0.03849629	99.90388968	3.85563215
0.80 – 0.85	4.60872535	0.04851130	100.47421637	4.87867809
0.80 – 0.86	4.60617662	0.05751586	100.26721653	5.82308327
0.80 – 0.87	4.60735626	0.06613718	100.43859702	6.67385709
0.80 – 0.88	4.60726963	0.07878499	100.52441041	8.04879429
0.80 – 0.89	4.61165033	0.08685758	101.03406735	8.93665773
0.80 – 0.90	4.61126455	0.09787492	101.09943897	10.09137874

Table 3.12: Copy Number Estimates for 1,000 Simulations, with $n = 8$ and $Z_0 = 100$

For both 50 and 100 initial copy numbers of the target, the model accurately estimates the initial copy number based upon all the information available from a PCR experiment. Based upon the results presented, the model should be accurate for all values of the initial copy number of the target, as long as the standards are close, but not necessarily the same, as the target copy number.

Range	Mean, Plate	Standard Deviation, Plate	Mean, Well	Standard Deviation, Well
0.80 – 0.81	0.00013788	0.00024184	0.00040535	0.00055851
0.80 – 0.82	0.00013314	0.00023359	0.00127754	0.00124425
0.80 – 0.83	0.00013423	0.00023212	0.00270684	0.00239778
0.80 – 0.84	0.00012609	0.00021775	0.00454465	0.00378027
0.80 – 0.85	0.00012033	0.00021941	0.00738794	0.00599078
0.80 – 0.86	0.00011973	0.00020921	0.01100876	0.00860520
0.80 – 0.87	0.00011861	0.00020629	0.01514946	0.01175692
0.80 – 0.88	0.00010677	0.00019147	0.01850699	0.01501609
0.80 – 0.89	0.00010885	0.00019771	0.02336718	0.01756214
0.80 – 0.90	0.00010221	0.00018126	0.02959798	0.02408302

Table 3.13: Covariance Parameters for Plate and Well Effects, for $n = 8$ and $Z_0 = 100$

3.3 Uses for Model

One use of the proposed model for PCR is for researchers who currently utilize a traditional PCR machine and do not wish to incur the added expense of a real-time PCR machine. The model allows the researcher to estimate the initial number of copies of the target sequence with a certain statistical accuracy. For further research, the distribution of the estimates for the initial copy number for the target sequence needs to be analyzed, and confidence intervals derived from the distribution determined.

REFERENCES

1. Chaing, P. W., Wei, W. L., Gibson, K., Bodmer, R., and Kurnit, D. M. A fluorescent quantitative PCR approach to map gene deletions in the Drosophila genome. Genetics, 153: 1313-1316, 1999.

2. Cohrs, R. J., Randall, J., Smith, J., Gilden, D. H., Dabrowski, C., Van Der Keyl, H., and Tal-Singer, R. Analysis of individual human trigeminal ganglia for latent herpes simplex virus type 1 and varicella-zoster virus nucleic acids using real-time PCR. J. Virol, 74: 11464-11471, 2000.

3. Costa, J. M., Pautas, C., Ernault, P., Foulet, F., Cordonnier, C., and Bretagne, S. Real-time PCR for diagnosis and follow-up of toxoplasma reactivation after allogenic stem cell transplantation using fluorescence resonance energy transfer hybridization probes. J. Clin Microbiol, 38: 2929-2932, 2000.

4. DeMuth, J. P., Weaver, D. A., Crawford, E. L., Jackson, C. M., and Willey, J. C. Loss of spr1 expression measurable by quantitative RT-PCR in human bronchogenic carcinoma cell lines. Am. J. Resp. Cell Mol., 19: 25-29, 1998.

5. Hayward, A. L., Oefner, P. J., Sabatini, S., Kainer, D. B., Hinojos, C. A., and Doris, P. A. Modeling and analysis of competitive RT-PCR. Nucleic Acids Res, 11: 2511-2518, 1998.

6. Higuchi, R., Fockler, C., Dollinger, G., and Watson, R. Kinetic PCR analysis: real-time monitoring of DNA amplification reactions. Bio-Technol, 11: 1026-1030, 1993.

7. Kawamoto, S., Ohnishi, T., Kita, H., Chisaka, O., and Okubo, K. Expression profiling by iAFLP: A PCR-based method for genome-wide gene expression profiling. Genome Res, 9: 1305-1312, 1999.

8. Kimura, H., Morita, M., Yabuta, Y., Kuzushima, K., Kato, K., Kojima, S., Matsuyama, T., and Morishima, T. Quantitative analysis of Epstein-Barr virus load by using a real-time PCR assay. J. Clin Microbiol., 39: 132-136, 1999.

9. Levinovitz, A. W., and Ringertz, N., eds. The Noble Prize: The first 100 years. Imperial College Press & World Scientific Publishing Co. Ptc. Ltd., 2001.

10. Lewin, S. R., Vesanen, M., Kostrikis, L., Hurley, A., Duran, M., Zhang, L., Ho, D. D., and Markowitz, M. Use of real-time PCR and molecular beacons to detect virus replication in human immunodeficiency virus type 1-infected individuals on prolonged effective antiretroviral therapy. J. Virol., 73: 6099-6102, 1999.

11. Liu, Q., Thorland, E. C., Heit, J. A., and Sommer, S. S. Overlapping PCR for bidirectional PCR amplification of specific alleles: a rapid one-tube method for simultaneously differentiating homozygotes and heterozygotes. Genome Res., 7: 389-398, 1997.

12. Machida, U., Kami, M., Fukui, T., Kazuyama, Y., Kinoshita, M., Tanaka, Y., Kanda, Y., Ogawa, S., Honda, H., Chiba, S., Mitani, K., Muto, Y., Osumi, K., Kimura, S., and Hirai, H. Real-time automated PCR for early diagnosis and monitoring of cytomegalovirus infection after bone marrow transplantation. J. Clin Microbiol., 38: 2536-2542, 2000.

13. Mathieu-Daudé, F., Welsh, J., Vogt, T., and McClelland, M. DNA rehybridization during PCR: the 'C_ot effect' and its consequences. Nucleic Acids Res., 24: 2080-2086, 1996.

14. Mellado, B., Gutierrez, L., Castel, T., Colomer, D., Fontanillas, M., Castro, J., and Estapé J. Prognostic significance of the detection of circulating malignant cells by reverse transcriptase-polymerase chain reaction in long-term clinically disease-free melanoma patients. Clin. Cancer Res., 5: 1843-1848, 1999.

15. Meselson, M., and Stahl, F. W. The replication of DNA in *Escherichia coli*. Biology, 44: 671-682,1958.

16. Navidi, W., Arnhein, N., and Waterman, M. S. A multiple-tube approach for accurate genotyping of very small DNA samples by using PCR: statistical considerations. Am. J. Hum. Genet., 50: 347-359, 1992.

17. Oaks, J. L., McGuire, T. C., Ulibarri, C., and Crawford, T. B. Equine infectious anemia is found in tissue macrophages during subclinical infection. J. Virol., 72: 7263-7269, 1998.

18. Pahl, A., Kühlbrandt, U., Brune, K., Röllinghoff, M., and Gessner, A. Quantitative detection of *Borrelia burgdorferi* by real-time PCR. J. Clin. Microbiol., 37: 1958-1963, 1999.

19. Pfaffl, M. W. A new mathematical model for relative quantification in real-time RT-PCR. Nucleic Acids Res., 29: 2002-2007, 2001.

20. Quadroni, M., James, P., Dainese-Hatt, P., and Kertesz, M. Proteome mapping, mass spectrometric sequencing and reverse transcription-PCR for characterization of

the sulfate starvation-induced response in *Pseudomonas aeruginosa* PAO1. Eur. J. Biochem., 266: 986-996, 1999.

21. Rao, V. and Saunders, N. B. A rapid polymerase-chain-reaction-directed sequencing strategy using a thermostable DNA polymerase from *Thermus flavus*. Gene. 113: 17-23, 1992.

22. Ringel, M. D., Balducci-Silano, P. L., Anderson, J. S., Spencer, C. A., Silverman, J., Sparling, Y. H., Francis, G. L., Burman, K. D., Wartofsky, L., Ladenson, P. W., Levine, M. A., and Tuttle, R. M. Quantitative reverse-transcription-polymerase chain reaction of circulating thyroglobulin messenger ribonucleic acid for monitoring patients with thyroid carcinoma. J. Clin. Endocr. Metab., 84: 4037-4042, 1999.

23. Sherby, L. S., ed. The Who's Who of Nobel Prize Winners, 1901-2000. Oryx Press, 2002.

24. Sun, F. The polymerase chain reaction and branching processes. J. Comput. Biol., 2: 63-86, 1995.

25. Takano, T., Miyauchi, A., Matsuzuka, F., Liu, G., Higashiyama, T., Yokozawa, T., Kuma, K., and Amino, N. Preoperative diagnosis of medullary thyroid carcinoma by RT-PCR using RNA extracted from leftover cells within a needle used for fine needle aspiration biopsy. J. Clin. Endocr. Metab., 84: 951-955, 1999.

26. Thomas, R., McConnell, R., Whittacker, J., Kirkpatrick, P., Bradley, J., and Sandford, R. Identification of mutations in the repeated part of the autosomal dominant polycystic kidney disease type 1 gene, PKD1, by long-range PCR. Am. J. Hum. Genet., 65: 39-49, 1999.

27. Torres, M. J., Criado, A., Palomares, J. C., and Aznar, J. Use of real-time PCR and fluorimetry for rapid detection of rifampin and isoniazid resistance-associated mutations in *Mycobacterium tuberculosis*. J. Clin. Microbiol., 38: 3194-3199, 2000.

28. Vidyashankar, A. Estimating the initial copy number from a PCR experiment. Unpublished manuscript, University of Georgia, January 2002.

29. Waller, C. F., Dennebaum, G., Feldmann, C., and Lange, W. Long-template DNA polymerase chain reaction for the detection of the *bcr/abl* translocation in patients with chronic myelogenous leukemia. Clin. Cancer Res., 5: 4146-4151, 1999.

30. Wharton, R. Q., Jonas, S. K., Glover, C., Khan, Z. A. J., Klokouzas, A., Quinn, H., Henry, M., and Allen-Mersh, T. G. Increased detection of circulating tumor cells in the blood of colorectal carcinoma patients using two reverse transcription-PCR assays and multiple blood samples. Clin. Cancer Res., 5: 4158-4163, 1999.

31. White, T. J., Arnhein, N., and Erlich, H. A. The polymerase chain reaction. TIG, 5: 185-188, 1989.

32. Willey, J. C., Coy, E. L., Frampton, M. W., Torres, A., Apostolakos, M. J., Hoehn, G., Schuermann, W. H., Thilly, W. G., Olsen, D. E., Hammersley, J. R., Crespi, C. L., and Utell, M. J. Quantitative RT-PCR measurement of cytochromes p450 1A1, 1B1, and 2B7, microsomal epoxide hydrolase, and NADPH oxidoreductase expression in lung cells of smokers and nonsmokers. Am. J. Respir. Cell Mol., 17: 114-124, 1997.

33. Willey, J. C., Crawford, E. L., Jackson, C. M., Weaver, D. A., Hoban, J. C., Khuder, S. A., and DeMuth, J. P. Expression measurement of many genes simultaneously

by quantitative RT-PCR using standardized mixtures of competitive templates. Am. J. Respir. Cell Mol., 19: 6-17, 1998

APPENDI X A

RANDOM NUMBER GENERATOR HEADER FILE, C++

```
#ifndef _INC_MATH
#include <math.h>
#endif

#define PI 3.141592654
// The following are defined for the ran2 subfunction
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0-EPS)

double unifrand(long* idum);  // Function Prototype for unifrand
double gammrand(int ia, long* idum);  // Func. Prototype for gammrand
double exprand(int ia, long* idum); // Func. Prototype for exprand
double betarand(int ia, long *idum);  // Func. Prototype for betarand
double poissrand(double xm, long *idum);  // Func. Prototype for poissrand
double gammln(double xx);  // Func. Prototype for gammln
double normrand(double mean,double sd,long *idum); //Fun. Proto. for normrand

// ************THE RANDOM NUMBER GENERATORS******
// Function to compute uniform random numbers
// Taken from Numerical Recipes in C
double unifrand(long *idum) //ran2
{
        int j;
        long k;
        static long idum2=123456789;
        static long iy=0;
```

```
        static long iv[NTAB];
        double temp;

        if (*idum <= 0) {
                if (-(*idum) < 1) *idum=1;
                else *idum = -(*idum);
                idum2=(*idum);
                for (j=NTAB+7; j>=0; j--){
                        k=(*idum)/IQ1;
                        *idum=IA1*(*idum-k*IQ1)-k*IR1;
                        if (*idum < 0) *idum += IM1;
                        if (j < NTAB) iv[j] = *idum;
                }
                iy=iv[0];
        }
        k=(*idum)/IQ1;
        *idum=IA1*(*idum-k*IQ1)-k*IR1;
        if (*idum < 0) *idum += IM1;
        k=idum2/IQ2;
        idum2=IA2*(idum2-k*IQ2)-k*IR2;
        if (idum2 < 0) idum2 += IM2;
        j=iy/NDIV;
        iy=iv[j]-idum2;
        iv[j]=*idum;
        if (iy < 1) iy += IMM1;
        if ((temp=AM*iy) > RNMX) return RNMX;
        else return temp;
}

// BETA RANDOM NUMBERS
double betarand(int ia, long *idum)
{
   double gammrand(int ia, long *idum);
        double r1, r2;
        r1=gammrand(ia,idum);
        r2=gammrand(ia,idum);
        return r1/(r1+r2);
}

//  GAMMA RANDOM NUMBERS
double gammrand(int ia, long *idum)
{
   double unifrand(long *idum);
//    void nrerror(char error_text[]);
   int j;
```

```
    double am,e,s,v1,v2,x,y;

//    if(ia<1) nrerror("Error in routine gamdev");
    if(ia<6){
      x=1.0;
      for(j=1;j<=ia;j++) x*=unifrand(idum);
      x=-log(x);
    } else {
       do {
         do {
           do {
             v1=unifrand(idum);
             v2=2.0*unifrand(idum)-1.0;
           } while (v1*v1+v2*v2 > 1.0);
           y=v2/v1;
           am=ia-1;
           s=sqrt(2.0*am+1.0);
           x=s*y+am;
         } while (x <= 0.0);
         e=(1.0+y*y)*exp(am*log(x/am)-s*y);
       } while (unifrand(idum) > e);
    }
    return x;
}

// EXPONENTIAL RANDOM NUMBERS
double exprand(int ia, long *idum){
          double unifrand(long *idum);
          double r;

          do
            r=unifrand(idum);
          while (r==0.0);
          return -ia*log(r);
}

// log gamma function that is needed for the Poison random number generator
double gammln(double xx){
          double x,y,tmp,ser;
          static double cof[6]={76.18009172947146,-86.50532032941677,
                    24.01409824083091,-1.231739572450155,
                    0.1208650973866179e-2,-0.5395239384953e-5};
          int j;
          y=x=xx;
          tmp=x+5.5;
```

```
        tmp -= (x+0.5)*log(tmp);
        ser=1.000000000190015;
        for (j=0; j<=5; j++) ser += cof[j]/++y;
        return -tmp+log(2.5066282746310005*ser/x);
}

// POISON RANDOM NUMBERS
double poissrand(double xm, long *idum){
        double gammln(double xx);
        double unifrand(long *idum);
        static double sq, alxm,g,oldm=(-1.0);
        double em,t,y;

        if (xm < 12.0) {
                if (xm != oldm) {
                        oldm=xm;
                        g=exp(-xm);
                }
                em=-1;
                t=1.0;
                do {
                        ++em;
                        t *= unifrand(idum);
                } while (t>g);
        } else {
                if (xm != oldm) {
                        oldm=xm;
                        sq=sqrt(2.0*xm);
                        alxm=log(xm);
                        g=xm*alxm-gammln(xm+1.0);
                }
                do {
                        do {
                                y=tan(PI*unifrand(idum));
                                em=sq*y+xm;
                        } while (em<0.0);
                        em=floor(em);
                        t=0.9*(1.0+y*y)*exp(em*alxm-gammln(em+1.0)-g);
                } while (unifrand(idum) > t);
        }
        return em;
}


// NORMAL RANDOM NUMBER GENERATOR
```

```
double normrand(double mean,double sd,long *idum){
	double unifrand(long *idum);
	static int iset=0;
	static double gset;
	double fac,rsq,v1,v2;

	if(*idum < 0) iset=0;
	if(iset==0){
		do {
			v1=2.0*unifrand(idum)-1.0;
			v2=2.0*unifrand(idum)-1.0;
			rsq=v1*v1+v2*v2;
		}
		while (rsq >= 1.0 || rsq == 0.0);
		fac=sqrt(-2.0*log(rsq)/rsq);
		gset=v1*fac;
		iset=1;
		return (v2*fac)*sd+mean;
	} else {
		iset=0;
		return (gset)*sd+mean;
	}
}
//******END OF RANDOM NUMBER GENERATORS*******
```

APPENDIX B

EXTINCTION PROBABILITY PROGRAM, C++

```
#include <iostream>
#include <iomanip>
#include <fstream>
#ifndef _INC_MATH
#include <math.h>
#endif
#include "myrand.h"  // see appendix A
using namespace std;

int main(void)                    // Main Program for Simulations
{
   long dum=2968746347L;         // The seed for random number generator
   long *pdum = &dum;            // pointer to seed dum

   ofstream outfile("extinct2.txt");  // output file

   for(int n=22; n<30; n++){          // number of gen's
         double p=0.40;              // prob of failure { no repro by a parent }
            double simext=0.0;  // avg ext prob over all sim's for n and p
            for(int sim=0; sim<100; sim++){ // 1000 sim's for ext. prob
               int Y[36]={0}; Y[0]=2;    // init parent gen's. Start gen has 2 parents
                  double extprob=0.0;          // init extinction prob for this sim
                   int total=0;               // init'g # of parents in pop
                   for(int j=0; j<n; j++){     // cycles through parent gen's
                        for(int k=0; k<Y[j]; k++){    // cycles through parents in gen
                             int G=(log(1-unifrand(pdum)))/(log(1-p));// Geom RV
                             if(G==0) {extprob+=1.0;}  // Add up #parents don't repro
                            Y[j+1]+=G;     // next gen's # is sum current gen's children
                        }
                        total+=Y[j];           // total parents in gen's o through n-1
                   }
                   extprob=extprob/total;      // (# parents not repro)/(# of parents)
                   simext+=extprob;            // sum of extprob from each sim
         }
         simext=simext/100.0;             // avg extprob over sim's
         cout << n << " " << p << " " << 1-p << " " << simext << endl;   // check
         outfile << n << " " << p << " " << 1-p << " 0.6666666667 " << simext;
         outfile << endl;             // output data
```

```
        }
        return 0;
}
```

APPENDIX C

PCR MODEL SIMULATIONS, C++

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include "myrand.h"

using namespace std;

int branching(double p, int n, int X);  // Function prototype for branching

int main(void)   // Main Program for Simulations
{
    long dum=14965738L;   // The seed for random number generator
    long *pdum = &dum;  // pointer to seed dum

    ofstream outfile("z100_n8.txt");

    int n[4];   //initializing cycle numbers
    n[0]=8; n[1]=10; n[2]=12; n[3]=15; //setting cycle numbers
    int Z0=100, X1=40, X2=60, X3=80, X4=120, X5=140;

    for(int i=0; i<4; i++){
        for(int t=1; t<=10; t++){
            double prep=0;      // the width of prob interval
            prep=(.9-.8)-((t-1.0)/100.0);
            for(int tt=1; tt<=t; tt++){
                double RL=0.0;   //initializing lower limit of prob range
                double RU=0.0;   //initializing lower limit of prob range
                RL=.8+((tt-1.0)/100.0);  // Lower limit of prob range
                RU=RL+prep;              // Upper limit of prob range
                for(int sim=1; sim<=1000; sim++){   // # of sim's
                    for(int j=0; j<10; j++){      //cycle through prob's
                        double p[10]={0};  //prob of reproducing
                        double U=0;  // Unif RV
                        U = unifrand(pdum);
                        p[j] = prep*U+RL;   // p ~ Unif(RL, RU)
                    }
                    for(int plate=1; plate<=6; plate++){
                        int X[10]; //where X.,1 are the X1n's, etc
```

```
                        X[0] = branching(p[0],n[i],X1);    // Calling branching function
                        X[1] = branching(p[1],n[i],X2);
                        X[2] = branching(p[2],n[i],X3);
                        X[3] = branching(p[3],n[i],X4);
                        X[4] = branching(p[4],n[i],X5);
                        X[5] = branching(p[5],n[i],Z0);
                        X[6] = branching(p[6],n[i],Z0);
                        X[7] = branching(p[7],n[i],Z0);
                        X[8] = branching(p[8],n[i],Z0);
                        X[9] = branching(p[9],n[i],Z0);
                        outfile << Z0 << " " << n[i] << " " << RL;  //output data
                        outfile << " " << RU << " " << sim << " ";
                        outfile << plate << " 1 " << X1 << " ";
                        outfile << X[1] << " " << Z0 << " " << X[6] << endl;
                        outfile << Z0 << " " << n[i] << " " << RL;
                        outfile << " " << RU << " " << sim << " ";
                        outfile << plate << " 2 " << X2 << " ";
                        outfile << X[2] << " " << Z0 << " " << X[7] << endl;
                        outfile << Z0 << " " << n[i] << " " << RL;
                        outfile << " " << RU << " " << sim << " ";
                        outfile << plate << " 3 " << X3 << " ";
                        outfile<< X[3] << " " << Z0 << " " << X[8] << endl;
                        outfile << Z0 << " " << n[i] << " " << RL;
                        outfile << " " << RU << " " << sim << " ";
                        outfile << plate << " 4 " << X4 << " ";
                        outfile << X[4] << " " << Z0 << " " << X[9] << endl;
                        outfile << Z0 << " " << n[i] << " " << RL;
                        outfile << " " << RU << " " << sim << " ";
                        outfile << plate << " 5 " << X5 << " ";
                        outfile << X[5] << " " << Z0 << " " << X[10] << endl;
                    }
                }
            }
        }
    }
        return 0;
}

//branching process function
int branching(double p, int n, int X){
        int zi=X;
        int Z=X;
        double r1;
        long dum=14965738L;   //seed for rand numb gen
        long *pdum = &dum;  // pointer to seed dum
```

```
        for(int i=1; i<=n; i++){
                for(int j=1; j<=Z; j++){
                        r1=unifrand(pdum);
                        if(r1>0 && r1<=p){zi += 1;}
                }
                Z=zi;
        }
        return Z;
}
```

APPENDIX D

SAS MACRO FOR ANALYZING PCR

```
%macro branch(ds,rlkeep,rukeep,filename,title);

/* INPUTTING DATA AND CREATING RESPONSE AND EXPLAN VARIABLES */
data &ds;
  infile "&ds..txt"; *delimiter='   ';
  input znot n rl ru sim plate well inistan finstan initar fintar;
  response=log(fintar/finstan);
  explan=log(inistan);
run;
data &ds;
  set &ds;
  where rl=&rlkeep and ru=&rukeep;
run;

ods output SolutionF=nout(keep=sim Effect Estimate);
ods output CovParms=noutC(keep=sim CovParm Estimate);

/* RUNNING PROC MIXED FOR EACH SIM */
proc mixed data=&ds noclprint noinfo noitprint noprofile;
   by sim;
   class well plate;
   model response = explan / solution;
   random well plate;
run;
ods listing;

proc sort data=nout; by sim effect; run;
proc transpose data=nout out=nout;
  by sim;
  var estimate;
  id effect;
run;

/*  CREATING DATASET OF ESTIMATES */
data nout;
  set nout;
  target=intercept/-explan;
  target2=exp(target);
  keep sim intercept explan target target2;
run;

/*  CREATING DATASET OF COVARIANCES */
proc sort data=noutC; by sim; run;
proc transpose data=noutC out=noutC;
  by sim;
  var estimate;
```

```
  id CovParm;
run;

ods rtf file="&filename..rtf" style=minimal;

title1 "&title";
title2 'Summary';
/* FINDING MEAN OF ESTIMATES FROM SIM'S */
proc means data=nout maxdec=8 n mean std min max clm;
  var intercept explan target target2;
run;
title3 'Covariance Parameter';
proc means data=noutC maxdec=8 n mean std min max clm;
  var well plate;
run;
title1;
title2;
title3;

ods rtf close;
ods output close;
ods listing close;

proc export data=nout outfile="&filename.sum.csv" dbms=csv replace;
proc export data=noutC outfile="&filename.cov.csv" dbms=csv replace;
run;

%mend branch;


/* EXAMPLE OF CALLING MACRO */
%branch(n12z100,.80,.90,n12z100_80_90,'Parameters are n=12 z=100 and
           p~uniform(.80,.90)');



endsas;
```