Mathematical Model for Time to Neuronal Apoptosis Due to Accrual of DNA DSBs

C. Mohanakumar[5], A. E. Offer[4], J. Rodriguez[2], B. Espinoza[1], V. Moreno[1], F. Nazari[1], C. Castillo-Garsow[3], D. Bichara[1]

[1] Arizona State University, Tempe, AZ, USA

[2] California State University, Channel Islands, CA, USA

[3] Eastern Washington University, Cheney, WA, USA

[4] Texas Tech University, Lubbock, TX, USA

[5] University of Florida, Gainesville, FL, USA

Abstract

We propose a mechanism to explain neuronal aging by tracking the number of non-transient DNA double-strand breaks (DSBs) and repairs over time that may lead to apoptosis. Neuronal apoptosis depends on the amount of space between DSBs as well as time. We derive three models to track the effect of neurodegeneration: a system of autonomous Ordinary Differential Equations (ODEs), a probability model to track the spatial requirement, and a stochastic model that incorporates both the ODE temporal dynamics and a spatial probability model. Using these models, we estimate a distribution for the lifespan of a neuron and explore the effect of parameters on time to death. We identify three possible causes of premature neuronal apoptosis: problems with coding critical repair proteins, issues with the neuron detecting DSBs, and issues with the neuron responding to DSBs.

1 Introduction

1.1 Neurodegeneration and DNA Double-Strand Breaks

Neurodegeneration, an umbrella term for a wide range of conditions that negatively affect the neurons in the human brain [10], is an increasingly prevalent and important issue due to the gradual increase of the average lifespan in the US population over the past 100 years. An increasing amount of evidence points to the age-related accrual of DNA breakage as a culprit for this issue, since DNA breaks can cause neuronal apoptosis. It is established that repair mechanisms exist to reconnect and repair these breakages, but these repair processes can degrade over time, leading to potential neurodegeneration [6,11]. This is especially true in the case of potentially lethal double-strand breaks (DSBs), the principal molecular lesions of biological relevance.

In the United States, senior citizens are a growing population. By 2050, the US Census calculated that over 800,000 people will reach the age 100 [8]. In 2010, over 83,000 people in the United States died of Alzheimer's Disease [1]. Due to the growing proportion of citizens over the age of sixty-five, the prevalence of Alzheimer's Disease will continue to grow [1]. Other age-linked neurodegenerative diseases, such as Parkinson's Disease, Prion Disease, Motor Neurone Disease, Huntington's Disease, Spinocerebellar Ataxia, and Spinal Muscular Atrophy, are all undergoing a similar increase. Currently there are no treatments for neural aging. This is because the mechanism by which neural aging occurs is not fully understood [9]. It is necessary to delineate the mechanism of neural aging and the neuronal progression to apoptosis. This will lead to future research into a treatment for age-related neurodegeneration.

The paper is organized as follows: the assumptions inherent in our model, followed by a description of our methodology and construction of the models, summary of our results, and a discussion of the results of our stochastic simulations.

1.2 Model Overview

We propose an ODE mathematical model with the independent variable of time. By the nature of this time-dependent ODE system, the ODEs do not track the spatial component of neuronal apoptosis–that is, the condition of a fragment of 20 base pairs or less breaking off in order to trigger neuronal apoptosis. Thus, what we may glean from this ODE model is not necessarily time to neuronal apoptosis, but rather time to a certain number of breakages or incorrect repairs. In addition, the ODE rates of repair and breaks help in the creation of the stochastic model, which shall be described later in this section.

In order to incorporate the spatial aspect of neuronal apoptosis, we create a probabilistic branching process in which we uniformly drop breaks on a continuous DNA strand. Using this probability model, we determine the theoretical LD50 (number of DSBs necessary to create a 50% chance of neuronal apoptosis) of a neuron. Due to the nature of biological research, the LD50 of DNA breakage will provide bountiful information for the "toxicity" of the DNA breaks. This is because the number of DSBs for which probability of death equals probability of survival gives us more information about neurons in general than an exact number of breaks at death for a specific cell. More specifically, this value gives us a threshold for neurodegeneration based on the accrual of DSBs.

The probability model incorporates our spatial aspect but not time. By combining ODE rates with the spatial process of the probabilistic model, we create a stochastic model to simulate multiple interactions modeled via Continuous-Time Markov Chain. In this way we simulate neuronal apoptosis and estimate the distribution of time of death for neurons. This distribution gives us a good idea of how DSBs affect neuronal

apoptosis based on our assumptions.

In addition, the stochastic model will be able to provide us with a numerical LD50, simply by running several simulations of neuronal apoptosis and calculating the median amount of breaks necessary to create neuronal apoptosis. This numerical LD50 may then be compared to the theoretical LD50.

Finally, we use the ODE model, which has the advantage of calculating time to a certain number of breaks, to compare the average time necessary to reach both the theoretical and numerical LD50s. We may then observe the impact of the parameters in our ODE model on the average time to LD50. This allows us to explore the parameter space more efficiently than with simulations and statistics, identifying parameter ranges of interest for detailed statistical analysis in the stochastic model.

2 Model Description

2.1 Biology of DNA DSBs

DNA double-strand breaks (DSBs) can be harmful. However, there is also evidence that DNA breakage is a crucial step in the learning process, storage of memory, and other natural processes in the human brain [6, 7, 11]. For example, DNA DSBs occur in the promoter regions of certain early response genes and are essential to expressing these genes, whose primary purpose is memory formation [6]. These types of activity-induced DSBs are called transient breaks, because when the body creates them, it repairs them almost immediately [6]. Harmful breaks (non-transient breaks) can be caused by many different factors, from products of cellular metabolism to pollutants, chemicals, and radiation [7]. These DSBs are more complex than transient breaks, so they tend to take more time repair. Mathematical models of the DNA breakage process, the underlying assumptions, and the biological phenomena associated with it have been far and few between. In this paper, we derive three mathematical models to better understand neurononal apoptosis in terms of non-transient DNA DSBs. The rate of repair of DNA DSBs decreases with age [3, 7, 11, 14], so these mathematical models will be focusing on the rate of repair of DSBs.

It has been verified that the method by which neurons repair DSBs is non-homologous end joining (NHEJ). This is the only process by which DSBs can be repaired in most neurons due to the fact that very few neurons replicate [7]. When a DSB occurs, a protein kinase, ataxia telangiectasia mutated (ATM), is called to the cite of the break to begin the process of NHEJ. ATM is namely responsible for the phosphorylation of H2AX which becomes γ-H2AX, a biomarker that allows lab scientists to detect DSBs [4]. The phosphorylated biomarker γ-H2AX is then responsible for recruiting other proteins required to repair the break, such as the Ku70/80 heterodimer, DNA-PKcs, Artemis, Polymerase $\lambda - \mu$, and the XRCC4/XLF/Ligase IV complex

[13]. To form a stable complex of Ku, approximately 20 base pairs (b.p.) are needed around the site of each DSB [5]. It is known that two breaks that occur within 20 b.p. of each other on a critical coding region will result in a break-off that cannot be repaired. This causes neuronal apoptosis, and this idea is incorporated into both the upcoming stochastic and probabilistic models.

It is important to note that the break-off of 20 b.p. is not the only way neurons undergo apoptosis due to DSBs. NHEJ is sometimes an error-prone process. In other words, although NHEJ repairs DNA DSBs, it will sometimes create a mutation (wrong repair) in the area of repair [2]. Sometimes there is a nonsense deleterious mutation, a mutation that stops the transcription of a protein early in the process. This results in a truncated protein that has lost some or all of its function. Without this essential protein, processes in the neuron stop, and in some cases the neuron dies. In order to focus our study on a single mechanism of apoptosis, our model does not include nonsense coding, but this will be discussed further in future research.

2.2 Assumptions

Returning to the model, we assume that the rates of both proper and improper repair are dependent on the number of DSBs and incorrect repairs of breakages in the DNA strand. We first assume that rate of repair is zero when the number of breaks and wrong repairs are zero, and then as DSBs increase, we assume that the efficiency of repair will increase, since the proteins needed for repair are already being produced. However, the proteins used during non-homologous end joining (NHEJ) are coded in the DNA they are attempting to repair. Thus, at some point, the accrual of DSBs and wrong repairs interfere in the creation of these essential proteins. At some point the number of DSBs and wrong repairs will start to negatively affect the neuron's ability to repair DSBs. This behavior is more thoroughly described in the description of $\alpha(B, W)$.

How quickly DNA DSBs repair depend on many factors, including the area that is broken, how close the break is to repair proteins, and how close the break is to other DSBs and DNA mutations. However, we assume that repair and breakage both occur randomly, since the DNA strand has millions of protein coding base pairs. As such, the probability of a wrong repair being broken again is close enough to zero that simplify the model by letting the wrongly repaired compartment be a source of inflow with zero outflow; this is true because of the continuous time Markov chain: there is no event. This is also built into our stochastic model for regular fixes through our relatively extreme parameter values.

As stated before, DSBs can either be transient or non-transient. Since transient breaks are a part of the transcription process in many cases, they are immediately recognized and repaired by the neuron [6]. Thus, we will assume that all of our models are only looking at non-transient DSBs.

3 Methods

3.1 ODE Model Framework

We consider the DNA base pairs composed of unbroken (U), broken (B), and wrongly repaired base pairs (W). Our proposed model describes the dynamics of DSBs in a single neuron and the model of ordinary differential equations is given by:

$$\frac{dU}{dt} = -bU + pB\alpha(B,W) \quad (1)$$

$$\frac{dB}{dt} = bU - B\alpha(B,W) \quad (2)$$

$$\frac{dW}{dt} = (1-p)B\alpha(B,W) \quad (3)$$

where, the total number of base pairs, $N = U + B + W$.

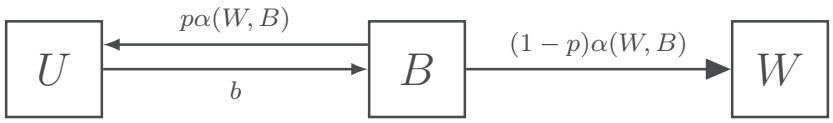

Figure 1: Movement and rates of change between the three classes, unbroken b.p. (U), broken b.p. (B), and wrongly repaired b.p. (W).

For our model, we consider a population of N base pairs (b.p.). We begin with a healthy neuron, so all of $N = U + B + W$ is contained in U (unbroken b.p.). When a DSB occurs, a broken b.p will move from U to B. When a DSB is detected by the cell, it will attempt to fix the DSB and move it out of the B at a rate of $B\alpha(B,W)$. The cell will then either repair the DSB properly and move the broken b.p. back into U, or repair it improperly and move it into W (Figure 1). We summarize our state variables and model parameters in Table 1.

Table 1: Description of the parameters and state variables.

Variables & Parameters	Description	Units
U	Number of unbroken DNA base pair (b.p.) linkages	b.p.
B	Number of broken DNA base pair linkages (DSBs)	b.p.
W	Number of wrongly repaired DNA base pair linkages	b.p.
b	Per capita rate of (harmful) breakage	1/time
a	Number of breakages that most excites the neuron	b.p.
c	Highest per capita repair rate	1/time
$\alpha(B, W)$	Per capita rate of proper/improper repair	1/time
p	Proportion of broken DNA that become properly repaired	dimensionless

The per capita rate of repair equation, $\alpha(B, W)$, is a function of B and W because of our fourth assumption, which will monotonically increase at the beginning, reach a maximum, and then monotonically decrease to an asymptote at the x-axis (Figure 2). The graph $\alpha(B, 0)$ will start at zero, because when there are no breaks, the per capita rate of repair is zero. As the breaks increase at first, the rate of repair will increase due to an increase in efficiency, but eventually the breaks and wrong repairs will have a negative effect and cause the rate of repair to decease.

Our $\alpha(B, W)$ functions should have these conditions:

- $\alpha(B, W) > 0$, the per capita rate of repair can never be negative.
- $\alpha(0, 0) = 0$, when there are no breaks or wrong repairs, the rate of repair is zero.
- $\alpha(B, 0)$ is increasing when $B < a$ and decreasing when $B > a$.
- $\alpha(B, 0)$ has a maximum at (a, c)
- $\lim_{(B+W)\to\infty} \alpha(B, W) = 0$, as the number of DSBs and wrong repairs is increasing indefinitely, the rate of repair for a single DSB is less and less efficient.
- $\lim_{(B+W)\to\infty} B\alpha(B, W) = 0$, as the number of DSBs and wrong repairs is increasing indefinitely, the rate of repair for all of the DSBs in a neuron is less and less efficient.

We have three candidates for the $\alpha(B, W)$ function. The trancendental function is used primarily in simulations, while the rational functions are used for ODE analysis.

$$\alpha_1(B,W) = \frac{c}{a}Be^{1-(B+W)/a}, \tag{4}$$

$$\alpha_2(B,W) = \frac{\sigma\frac{c}{a}B}{1+\eta\frac{B+W}{a}+(\eta\frac{B+W}{a})^3}, \quad \sigma = \frac{1}{2}(3+2^{2/3}),\ \eta = \frac{1}{2^{1/3}}, \tag{5}$$

$$\alpha_3(B,W) = \frac{\sigma\frac{c}{a}B}{1+\eta\frac{B+W}{a}+(\eta\frac{B+W}{a})^4}, \quad \sigma = \frac{1}{3}(4+3^{3/4}),\ \eta = \frac{1}{3^{1/4}}, \tag{6}$$

where σ and η are normalization factors.

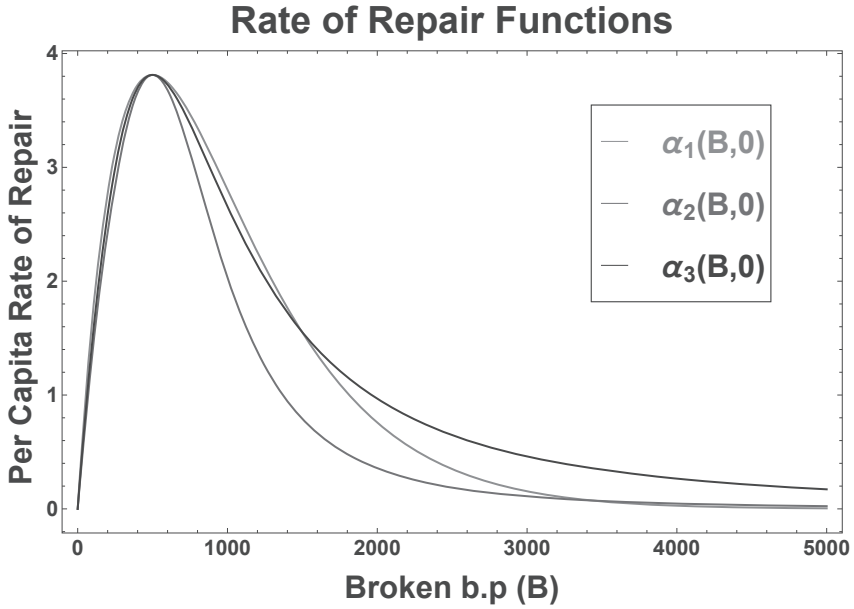

Figure 2: The equation for rate of proper and improper repair when there are no wrong repairs, $\alpha(B,0)$. Includes α_1, α_2, and α_3. ($\bar{a} = 500$, $\bar{c} = 3.812$)

3.2 Probability Model

In order for a neuron to undergo apoptosis, two breaks must happen within twenty base pairs (b.p) of each other [5]. Thus, in order to model apoptosis, we need a factor of not only time, but distance between DSBs, and to incorporate this into a model. While using the ODE model to analyze time, we create a separate probability model to analyze distance. The process begins with a number line from zero to one, representing the entire coding region of a single neuron compressed together to a continuous number line. We assume

that each break is placed at a random location distributed uniformly from zero to one. Let N equal the total amount of b.p in the coding region. Neuron death occurs when any two breaks are within $20/N = L$ of each other or an edge of the coding region $(0, 1)$. We are interested in the probability of surviving B breaks P(B). To do this, we develop a branching process probability model that reflects the probability of survival at stage B.

3.3 Stochastic Model

We combine the ODE and probability model into a single model by converting the ODE model to a continuous time Markov chain and combining it with the probability model to create a stochastic simulation. We create a code in MATLAB to simulate a neuron undergoing apoptosis due to DSBs. We use rates from our ODEs and break the model down into the three events that can occur in our model: a break, a repair, and a wrong repair (see Table 2). The time between each event is assumed to be exponentially distributed, since breaks and repairs themselves do not age. Every time a break occurs, we choose a location for the break. A location is defined by a number from 1 to 48 million, since this is the total number of base pairs in protein coding DNA. The locations of the breakages are recorded in a vector $\vec{B}$. If there is a wrong repair, one of the "locations" is removed from $\vec{B}$ and appended to the vector of wrong repairs, $\vec{W}$. If any of the locations are in $\vec{B}$ are within 20 b.p of each other, the simulation ends and we record time to apoptosis.

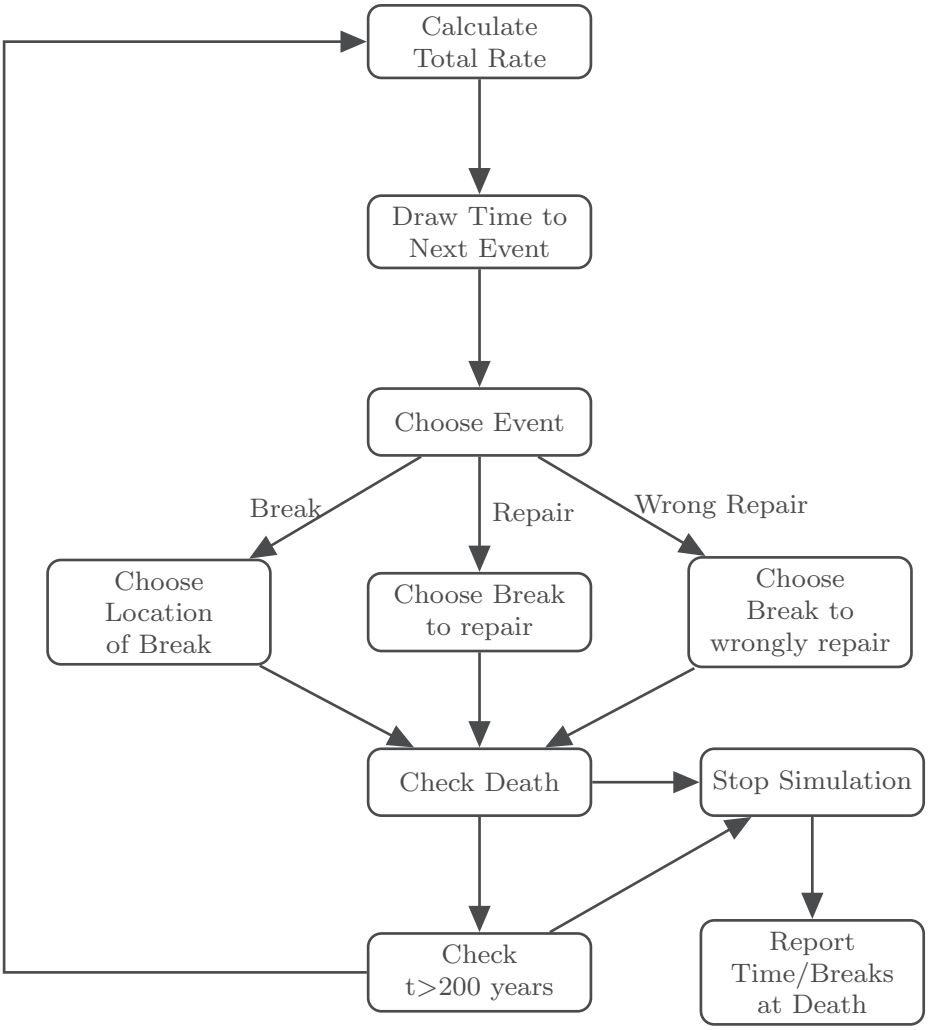

Figure 3: Outline of Stochastic Model

Table 2: List of events to be chosen using Continuous Time Markov Chain.

Event	Rate	Effect
Break (DSB)	b	$U = U - 1$, $B = B + 1$
Repair	$pB\alpha$	$U = U + 1$, $B = B - 1$
Wrong Repair	$(1-p)B\alpha$	$B = B - 1$, $W = W + 1$

3.4 Parameter Estimates

Our parameter b, is the per capita rate of harmful DSBs. The units of b are $\frac{1}{time}$ (this is a simplified version of $\frac{b.p}{time}$/b.p.). On average, 10-50 DSBs occur in mammalian cells everyday [7]. Since we are only looking

at non-transient breaks, we will choose the lowest of these numbers, 10 DSBs. Thus, b is $10\frac{b.p.}{day}$/3.2 billion b.p., since this rate is out of the total amount of human DNA. Thus, b becomes $\frac{10}{3.2billion}\frac{1}{time}$.

Our parameter c, is the maximum rate of repair when the neuron is most excited, with units $\frac{1}{time}$. A biochemical kinetic model for NHEJ repair of DSBs has been previously studied [13], and two pathways are analyzed. The first is a slow pathway (complex pathway) that we assume occurs in the repair of non-transient breaks. The second is a fast pathway, which we assume involves the repair of transient breaks. For the estimation of our parameter c the slow pathway is considered, as we are only focusing on the DSBs that do not get immediately repaired, non-transient DSBs. The reciprocals of the rate constants between each step in the repair mechanism are added to give a total time for a single repair. This yields the repair rate of $c = 3.812\frac{1}{day}$. Normal repair is assumed to occur at $c = 3.812$ where neuronal apoptosis due to DSBs does not occur. We interpret this rate to be the most healthy per-capita rate of repair for a neuron. Thus, a neuron's c can fluctuate down to $c = 0$, where there is no repair.

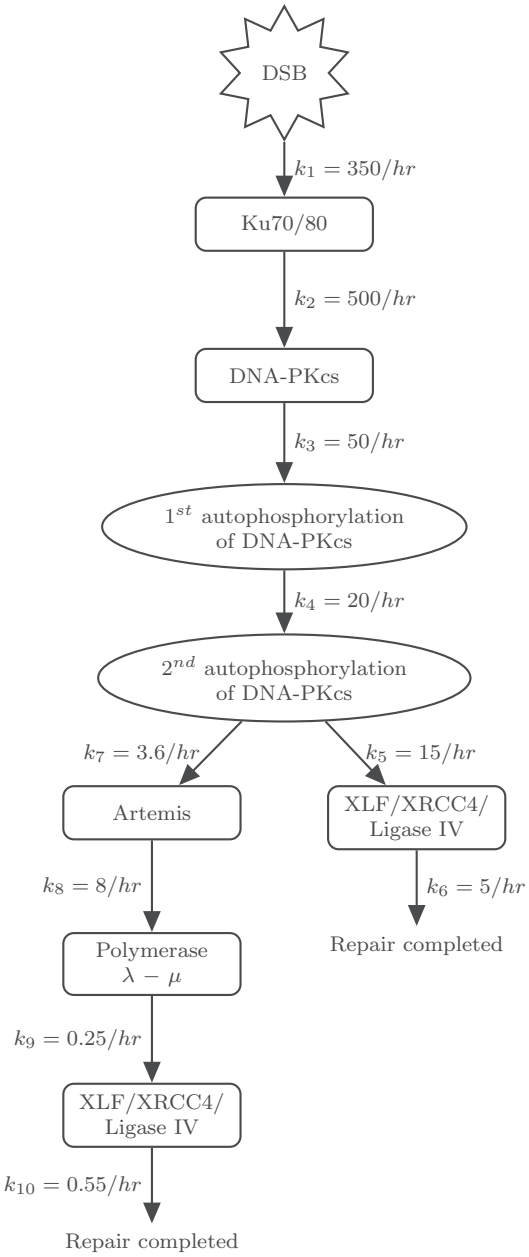

Figure 4: The process of NHEJ, going through all of the repair proteins to repair a DSB. The rate constants k_i from $i = 1$ to 10 for each protein were found and used to determine the most efficient time to repair a DSB. There are two pathways: the one of the left is the fast pathway that repairs transient DSBs, and the one of the right is the complex pathway that repairs non-transient DSBs. Summary of [13]

The proportion of broken DNA that is properly repaired is p. This is essentially the proportion of NHEJ repairs that happen correctly. On average, NHEJ is between 75% to 99.9% accurate in its repairs [2]. Thus, we picked p to be 87.5% or .875, with an interval of .75 to .999.

No data could be found on a, the number of breakages that most excite the neuron, or the number of DSBs at which the per capita repair rate is most effective. Thus, further analysis is required and discussed in Section 4.1. The parameter a is analyzed from a range of 10^0 to 10^{10}, with $c = 3.812$. The range of a from 10^2 to 10^3 is healthy (see Figure 5). Thus, we choose a to be 500, with an interval of 100 to 900 DSBs, but other intervals of a are also analyzed.

Table 3: Parameter estimations based on existing knowledge and data. (No existing data for a.)*

Parameters	Estimated Values	Intervals
$\bar{b}$	$\frac{10}{3{,}200{,}000{,}000}$	$(\frac{1}{3{,}200{,}000{,}000}, \frac{19}{3{,}200{,}000{,}000})$
$\bar{p}$	0.875	(0.75, 0.999)
$\bar{c}$	3.812	(0, 3.812)
$\bar{a}^*$	500	(100, 900)

4 Analysis

4.1 ODE Analysis

The rates α_2 and α_3 are used to find the equilibrium, with α_1 assumed to have similar properties. We then analyze the stability of the ODE model equilibrium. In both cases with α_2 and α_3, there is only one unique equilibrium: $(U, B, W) = (0, 0, N)$. This makes sense in terms of our model, because we assume that once there is a wrong repair, it cannot be broken again. However, this does not make sense biologically: If every base pair is wrongly repaired, the neuron cannot function. In reality, a neuron will likely undergo apoptosis long before even as much as half of its base pairs are wrongly repaired. This is not shown in the ODE model, because a death condition is not included in the model. In the stochastic simulation, the neuron undergoes apoptosis before many wrong repairs occur compared to the total number of coding base pairs.

In order to study the stability of the equilibrium, the ODE is reduced to only $\frac{dU}{dt}$ and $\frac{dB}{dt}$, since $N = U + B + W$. Using the Lyapunov function $V(B, U) = B + U$, we prove that the equilibrium $(U, B, W) = (0, 0, N)$, or $(U, W) = (0, 0)$ in the reduced system, is globally asymptotically stable.

Theorem 1. *In the ODE system, the sole unique equilibrium* $(U, B, W) = (0, 0, N)$ *is globally asymptotically stable.*

Proof. Consider the Lyapunov function, $V(B,U) = B + U$

then $V(B,U) \geq 0$ and $V(B,U) = 0$ if and only if $U = B = 0$

Notice that $\frac{dV}{dt} = \frac{dB}{dt} + \frac{dU}{dt}$ where $N = B + U + W$

$\frac{dV}{dt} = (p-1)B\alpha(B, N - B - U)$

then $\frac{dV}{dt} \leq 0$ since $\alpha(B, N - B - U) \geq 0$ and $p \leq 1$

Hence $\frac{dV}{dt} = 0$ if and only if $B = 0$ or $\alpha(B, N - B - U) = 0$

For $\alpha_1 = 0$, α_2, and α_3, $\alpha = 0$ if and only if $B = 0$

Thus, $\frac{dV}{dt} = 0$ if and only if $B = 0$

The largest invariant set where $\frac{dV}{dt} = 0$ is reduced to (0,0)

Hence by LaSalle's invariance principle the equilibrium (0,0) is globally asymptotically stable. □

It is necessary to analyze our choice of the parameter a, the number of DSBs at the most excited state of a neuron. As seen in Figure 5, when a is changed, there are different consequences for the neuron. When a is between twenty and one hundred, there are very few DSBs early in the neuron's life, but at some point in time before one hundred years the DSBs increase rapidly. In the range 10^5 to 10^9, the shape of DSB accumulation in the neuron is different. The number of breaks increase before leveling off. However, in the range of 100 to 900, the accumulation of DSBs stays low for the time period of 100 years. Thus, in this range a is considered a healthy parameter for the neuron. This is why we chose $a = 500$.

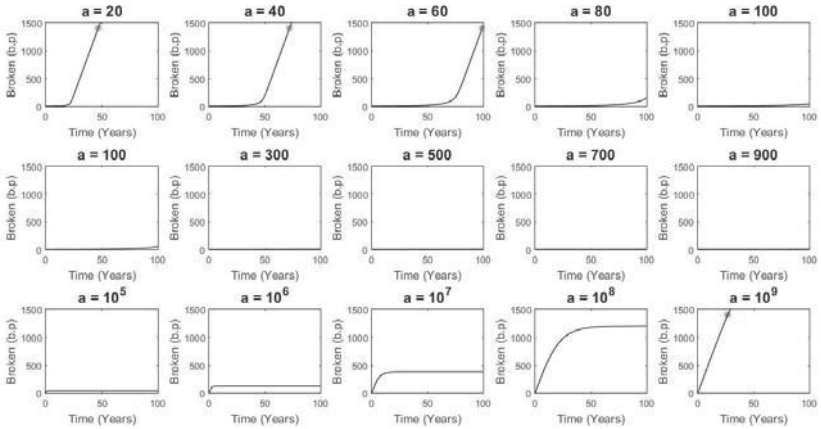

Figure 5: Accumulation of DSBs in a neuron over the course of 100 years, dependent on fluctuating parameter a. Three intervals are shown: $a = (20, 100)$, $a = (100, 900)$, and $a = (10^5, 10^9)$. The red points show where the thresholds 1400 DSBs(LD50) are hit. ($b = \frac{10}{3200000000}$, $p = 0.875$, $c = 3.812$, $N = 48000000$). More information on the estimation of LD50 can be found in Results.

4.2 Probability Analysis

In our probability model, we begin by creating a branching process. The process begins with a number line from zero to one, representing the entire coding region of a single neuron compressed together to a continuous number line. Let N equal the total amount of b.p in the coding region. We then erase regions in which a break would result in neuronal death — thus giving a probability of surviving the next break. At the beginning of the process, this would be represented by $20/N = L$ on both ends of our number line from zero to one; the remaining area is considered to be $p(1)$. We then proceed to add our first breakage, with a probability of success of $1 - 2L$. However, this break could be one of two types: chopping or splitting. Chopping is when the first break leaves region $p(1)$ as one region that can maintain another break in the next step and one region that cannot support another break. The region that cannot support another break is then erased just like the $20/N$ from both ends were erased at the beginning of our process. On the other hand, the first break will split $p(1)$ if it creates two regions that are still capable of supporting an additional break. We then create our branching process by considering the independent cases of split and chopping and adding additional breakages that can also cause either splitting and chopping.

This will be able to give us an explicit formula for the probability of survival of a neuron given N breakages. With this information, we are able to evaluate an average number of breakages necessary to create a fifty percent chance of death in the neuron, otherwise known as the LD50 of the neuron. We may then compare this to the median time necessary to achieve the LD50 that is constructed by the simulation.

Below is a flowchart of the first three generations of breakages. The symbols $p(B)^m$ signifies the m^{th} scenario of the B^{th} generation, while $p(B)_{sp}$ and $p(B)_{ch}$ represent the probability of whether the B^{th} generation break is a split or a chop.

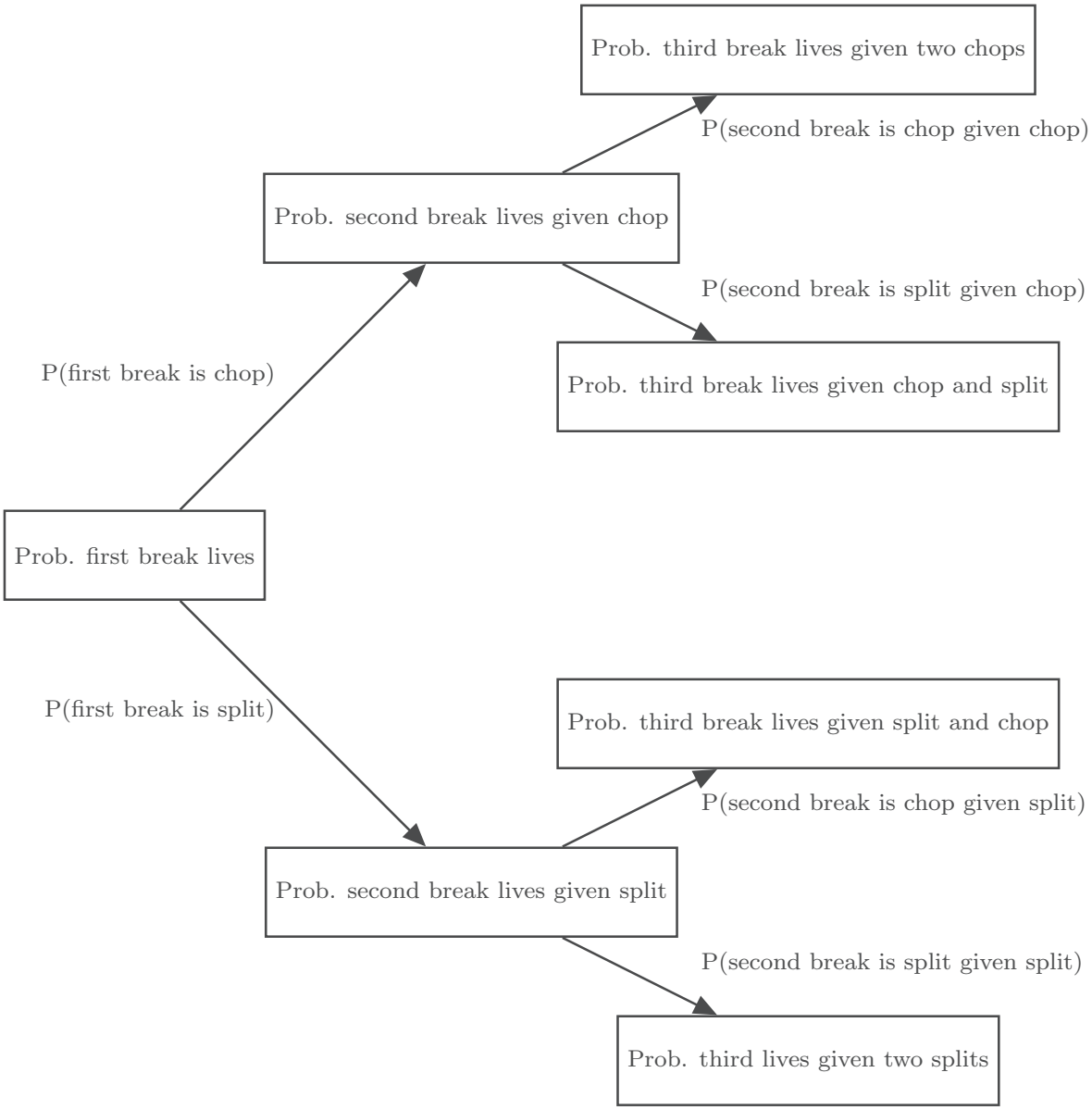

Figure 6: Branching process flowchart.

4.3 Distribution Fitting and Parameters

Upon running 10,000 simulations in our MATLAB code, we were able to create a histogram of times to neuronal death (see Figure 9). Next, we found the best fit for this histogram by using the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). The distributions we choose from were Gamma, Log–normal, Log–logistic, Weibull, and GEV. We find that that the Weibull distribution minimized both

the AIC and BIC, indicating that the Weibull distribution is the best fit. Further analysis is then based on the assumption that the Weibull is always the best fit. Next, we studied the affect of manipulating model parameters, b, p, a, and c, on the parameters of the Weibull, k and λ, instead of fitting a new histogram for each set of model parameter values.

The analysis of the parameter space is focused on the Weibull distribution.We can study the effect of the model parameters a, b, c, and p on time to death by studying the effect of these model parameters on the distributions parameters k and λ in the Weibull distribution.The parameter exploration is done via a uniform random sampling of the 4D parameter space, where a number in the interval of each parameter is chosen at random. These simulations are run 2,000 times for each parameter value, with a time limit of 200 years.

Additionally, we study the effect of low a with a one dimensional parameter exploration, where all other parameters were held at their base values, and a was tested in the interval $(10, 110)$ in eleven steps.

5 Results

5.1 Cumulative DSBs Over Time

In a time series analysis, broken b.p. (B) and wrongly repaired b.p. (W) steadily increase at a constant rate while unbroken, healthy b.p. (U) decrease at a steady rate. This is the simplest form of what is known about DSBs in relation to age: as a person ages, the number of DSBs in their DNA increases [3,7,11,12,14]. The stochastic version of the ODE uses the continuous-time Markov chain. In Figure 7, the stochastic ODE is represented by the multicolored plots. Keep in mind that this model does not incorporate any sort of spatial aspect, so each run of the neuron continues to accumulate DSBs and wrong repairs for the duration of 100 years. In reality, many neurons die out well before 100 years have passed. This is shown in the spatial stochastic model that incorporates distance between each break.

Two versions of the ODE models are shown in Figures 7 and 8: a neuron with a healthy per capita rate of repair ($c = 3.812$), and a neuron with an unhealthy per capita rate of repait ($c = 0$). Notice that in Figure 8 there are no wrong repairs, even in the stochastic version of the ODE. This is because the per capita rate of repair, c and the rate of repair itself is zero. This is an extreme case where there are no repairs being made in this neuron, so the DSBs are allowed to increase to N.

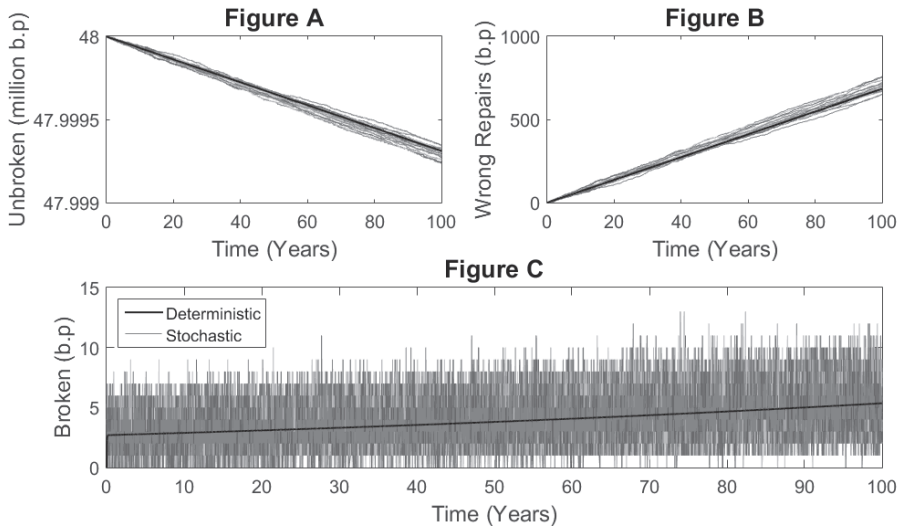

Figure 7: The deterministic ODE and 20 runs of the stochastic ODE in a healthy per capita rate of repair (when $c = 3.812$). (A) Number of unbroken/healthy base pairs (b.p.) in the protein coding region of a DNA strand over time of 100 years. (B) Number of wrongly repaired b.p in the protein coding region of a DNA strand over time of 100 years. (C) Number of broken b.p. (DSBs) in the protein coding region of a DNA strand over time of 100 years. ($\bar{a} = 500$, $\bar{b} = \frac{10}{3{,}200{,}000{,}000}$, $\bar{p} = 0.875$, $\bar{c} = 3.812$, $N = 48,000,000$)

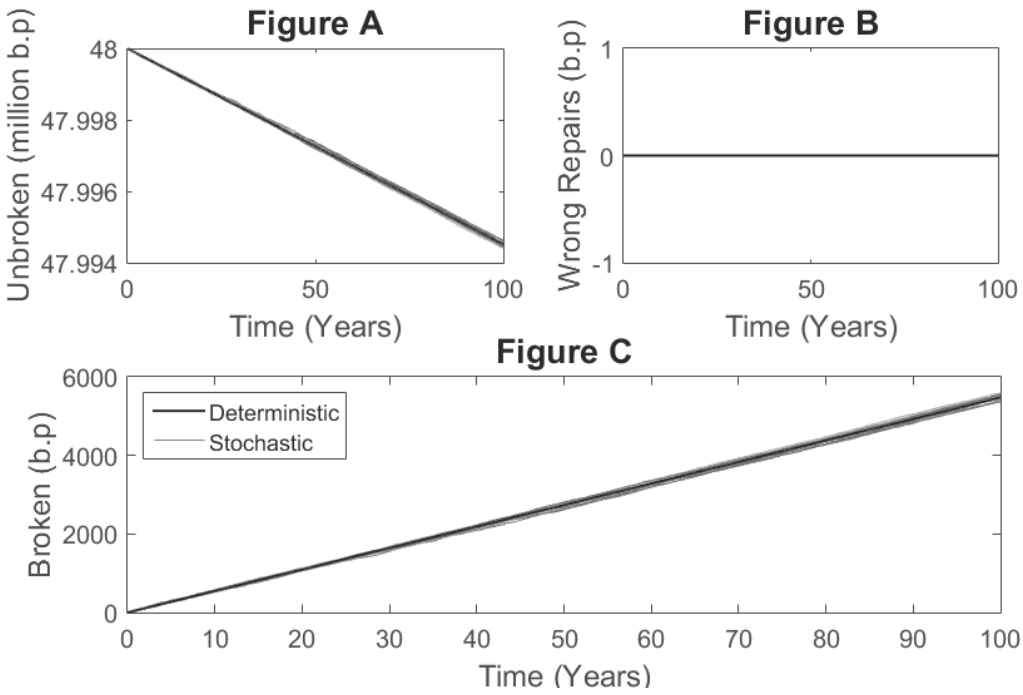

Figure 8: The deterministic ODE and 20 runs of the stochastic ODE in an unhealthy per capita rate of repair (when $c = 0$). (A) Number of unbroken/healthy base pairs (b.p.) in the protein coding region of a DNA strand over time of 100 years. (B) Number of wrongly repaired b.p. in the protein coding region of a DNA strand over time of 100 years. (C) Number of broken b.p. (DSBs) in the protein coding region of a DNA strand over time of 100 years. ($\bar{a} = 500$, $\bar{b} = \frac{10}{3{,}200{,}000{,}000}$, $\bar{p} = 0.875$, $c = 0$, $N = 48,000,000$)

5.2 Effects of Parameters on Average Time to Apoptosis

In the spatial stochastic ODE, neuronal apoptosis is incorporated and the distribution of times to apoptosis is analyzed. As stated above, we choose the Weibull distribution, because it minimized the AIC and BIC at the estimated parameter values ($\bar{a}$, $\bar{b}$, $\bar{p}$, and $c = 0$) (Figure 9). We chose $c = 0$ for our base distribution fit because at baseline parameter value $\bar{c} = 3.812$ all neurons lived over 200 years, making the distribution of time to death biologically meaningless.

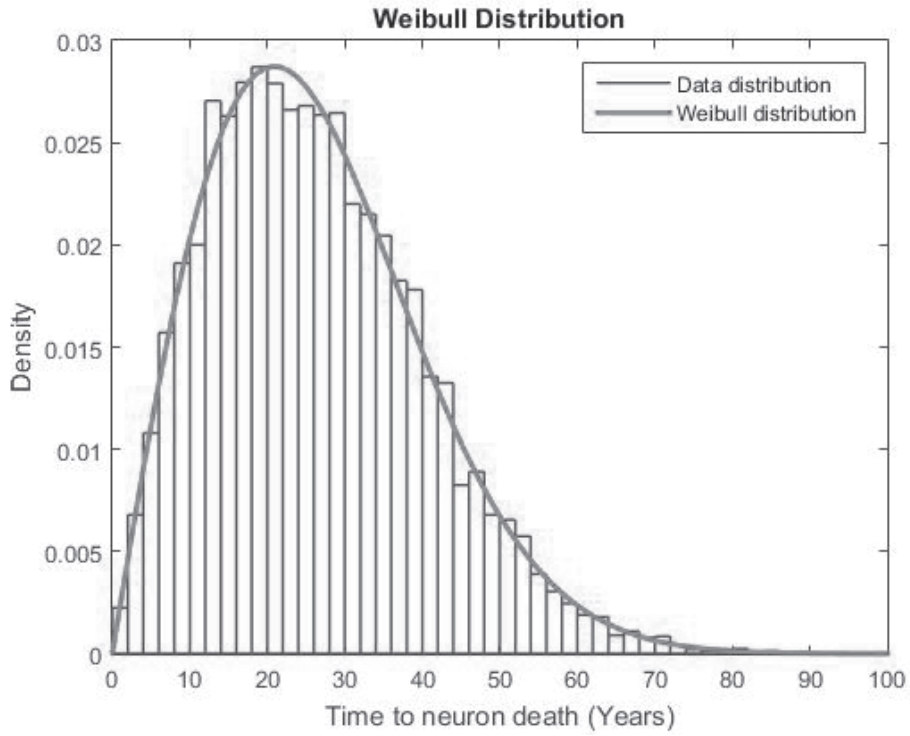

Figure 9: Distribution of the years it takes for a neuron with an unhealthy per capita rate of repair (c) to undergo apoptosis after 10,000 simulations. This figures shows the Weibull distribution as the best fit. ($\bar{a} = 500$, $\bar{b} = \frac{10}{3{,}200{,}000{,}000}$, $c = 0$, $\bar{p} = 0.875$, $N = 48,000,000$)

Table 4: Parameter estimations for Weibull distribution of times to neuronal apoptosis with base estimations and an unhealthy per capita rate of repair, c. ($\bar{a} = 500$, $\bar{b} = \frac{10}{3{,}200{,}000{,}000}$, $c = 0$, $\bar{p} = 0.875$)

Parameters	Estimation	Standard Error	CI min	CI max
λ	29.7378	0.157533	29.4307	30.0482
k	1.9878	0.0154924	1.9577	2.0184

We ran the spatial stochastic model in MATLAB 10,000 times to simulate a neuron dying with a distribution of times in years. The Weibull distribution is fitted to the distribution of times and is used to find k and λ (Figure 9). When there is no repair ($c = 0$) neuronal apoptosis occurs due to DSBs, and the time to apoptosis is distributed Weibull with parameters $\lambda = 29.7378$, $k = 1.9878$ (Table 4).

Further parameter exploration involves the effect of a,b,c,and p, on the parameters k and λ of the Weibull. Although we have not yet completed a full parameter exploration of the simulation, we can give an example: the effect of b in the case of low c. In a person with healthy a ($\bar{a} = 500$), the cell is quite robust, with the breaks accumulating beyond LD50 in the person's lifetime only if the rate of complex repair c falls below

approximately 5×10^{-4}. By choosing a c below this threshold, we examine how the rate of occurrence of non-transient breaks (those needing complex repairs) affects time to neuronal apoptosis. If such a person had a severely impaired ability to repair ($c = 0.0000216$), then time to neuronal apoptosis is Weibull distributed with parameters λ and k, where λ depends on b (Figure 10 and Table 5). The parameter k does not depend on b, where the mean and standard deviation of k are 1.980147 and 0.03819209, respectively. For ease of interpretation, we also examined the effect of b on mean and variance of time to neuron death, although the data is not Normal distributed (Figures 11, 12 and Tables 6, 7).

We then examine how rate of occurrence of damage affects the lifespan of a neuron in a person susceptible to neurodegenerative diseases, in particular when the parameter a is low (in $[10, 50]$ and c is normal ($c = 3.812$). Our preliminary results indicate that both, λ and k linearly depend on a within the range of interest (Figures 13, 15 and Tables 8, 10). The effect of a on the mean of time to neruon death was also analyzed (Figure 14, Table 9). An increase in the number of breakages when the per capita rate is the fastest will increase the lifespan of the neuron. Note that these results cannot be extrapolated to other values of a outside the interval [10,50]. For more information on the effect of other values of a, see LD50 results below.

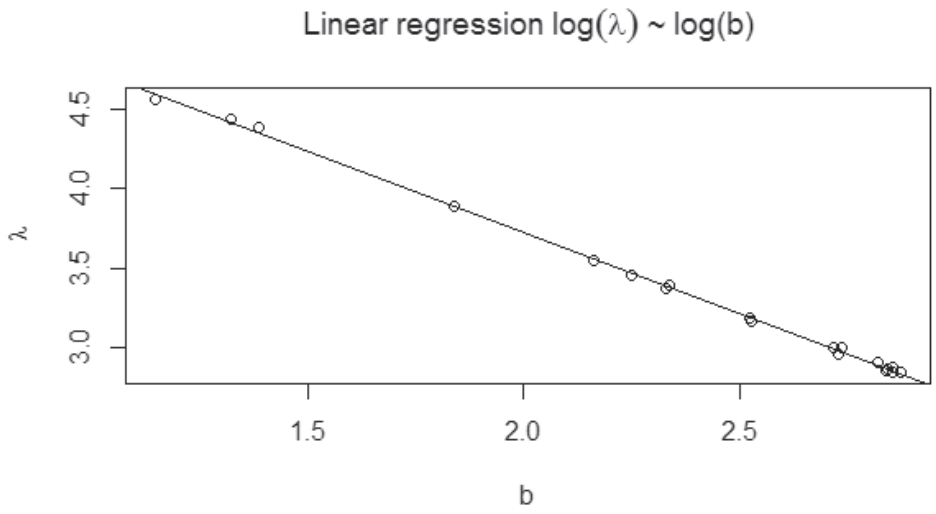

Figure 10: Linear regression with $log(\lambda) \sim 5.769434 - 1.021114 log(b)$

Table 5: Coefficients and R-squared values for linear regression of $log(\lambda) \sim log(b)$

	Estimation	Standard Error	t value	Pr($>|t|$)
intercept	5.769434	0.020158	286.2	<2e-16
$log(b)$	-1.021114	0.008283	-123.3	<2e-16

Multiple R-squared: 0.9989, Adjusted R-squared: 0.9988

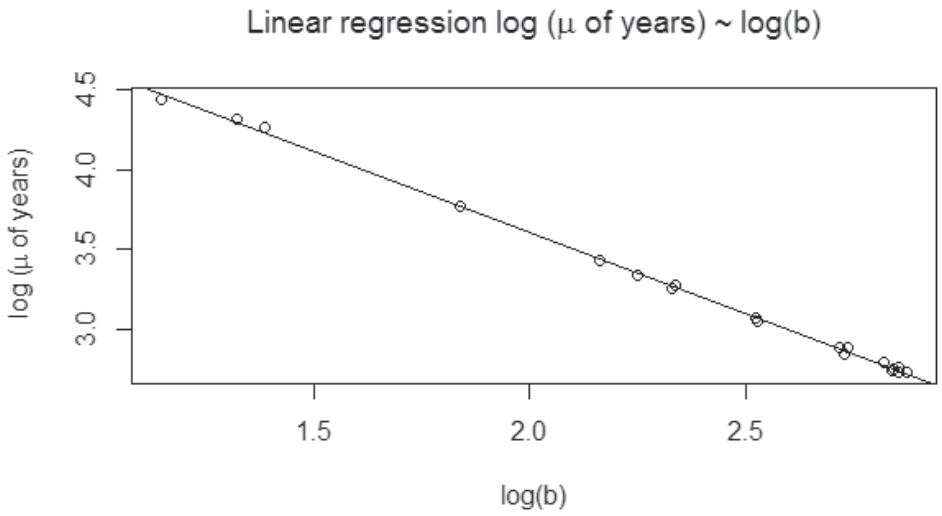

Figure 11: Linear regression with $log(\mu \text{ of years}) \sim 5.650154 - 1.021666 log(b)$.

Table 6: Coefficients and R-squared values for linear regression of $log(\sigma \text{ of years}) \sim log(b)$

	Estimation	Standard Error	t value	$\Pr(>\|t\|)$
intercept	5.650154	0.020314	278.1	$<2\times10^{-16}$
$log(b)$	-1.021666	0.008347	-122.4	$<2\times10^{-16}$

Multiple R-squared: 0.9989, Adjusted R-squared: 0.9988

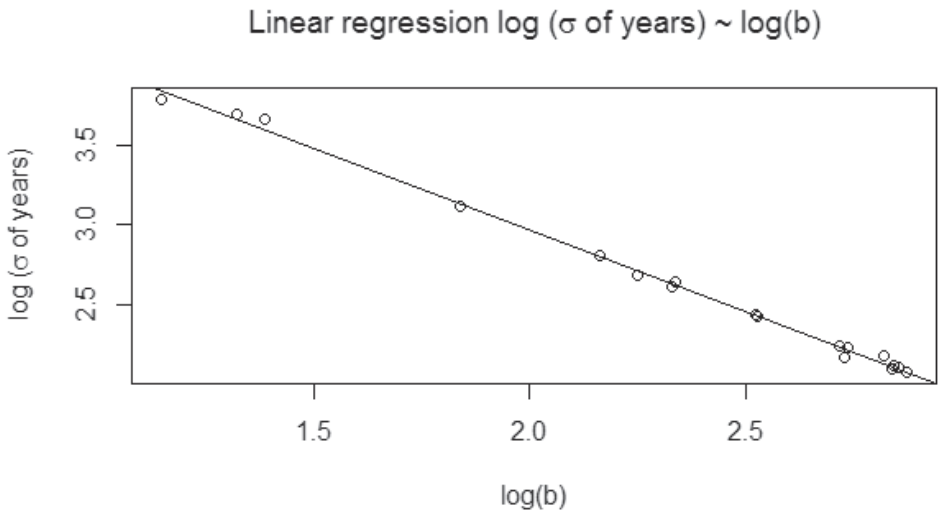

Figure 12: Linear regression with $log(\sigma \text{ of years} \approx 5.02856 - 1.02907 log(b)$

Table 7: Coefficients and R-squared values for linear regression of $log(\sigma \text{ of years}) \sim log(b)$

	Estimation	Standard Error	t value	$\Pr(>\|t\|)$
intercept	5.02856	0.03152	159.53	$<2\times10^{-16}$
$log(b)$	-1.02907	0.01295	-79.45	$<2\times10^{-16}$

Multiple R-squared: 0.9973, Adjusted R-squared: 0.9972

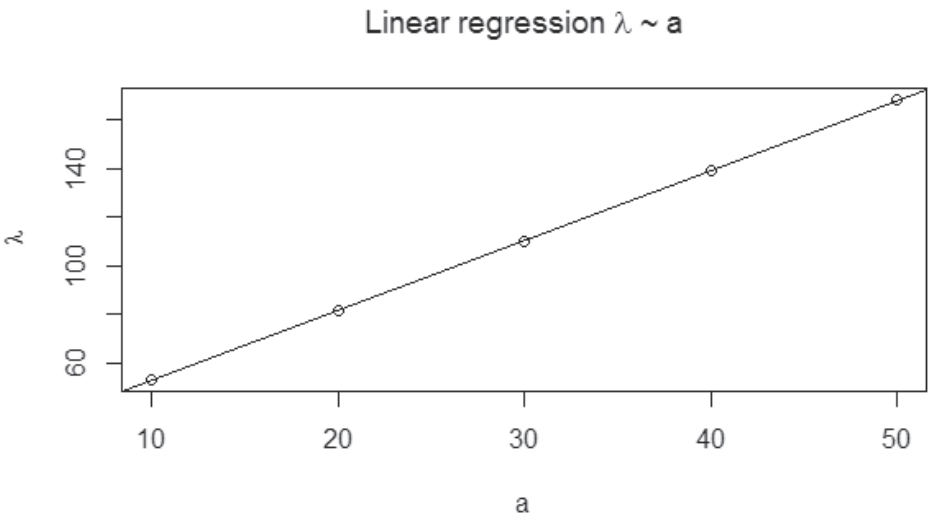

Figure 13: Linear regression with $\lambda \approx 23.71320 + 2.89096a$

Table 8: Coefficients and R-squared values for linear regression of $\lambda \sim a$

	Estimation	Standard Error	t value	$\Pr(>\|t\|)$
intercept	23.71320	0.48039	49.36	1.83×10^{-05}
a	2.89096	0.01448	199.59	2.77×10^{-07}

Multiple R-squared: 0.9999, Adjusted R-squared: 0.9999

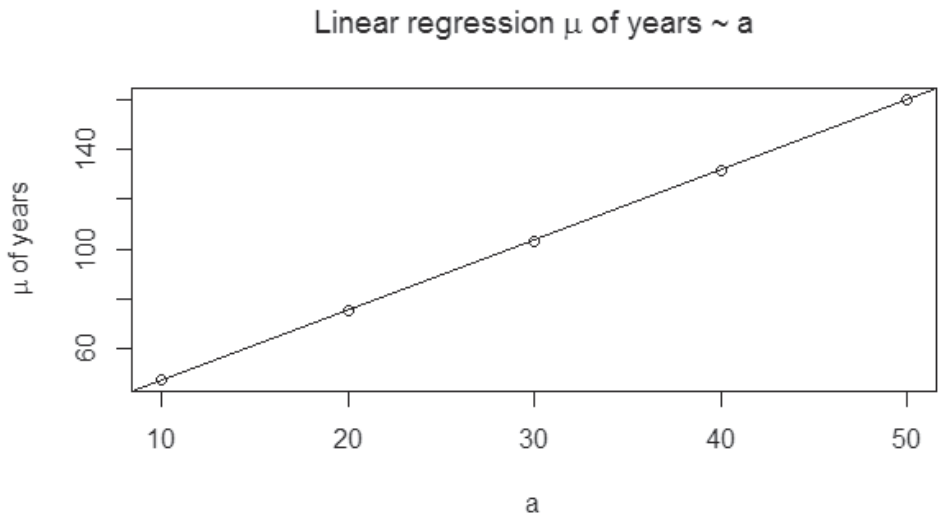

Figure 14: Linear regression with μ of years $\approx 19.39900 + 2.80286a$

Table 9: Coefficients and R-squared values for linear regression of μ of years $\sim a$

	Estimation	Standard Error	t value	$\Pr(>\|t\|)$
intercept	19.39900	0.48984	39.6	3.54×10^{-05}
a	2.80286	0.01477	189.8	3.23×10^{-07}

Multiple R-squared: 0.9999, Adjusted R-squared: 0.9999

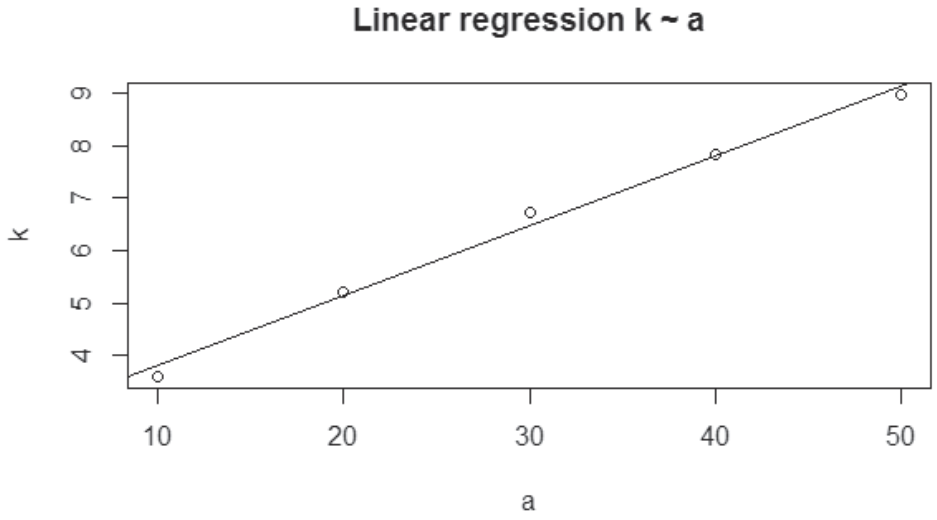

Figure 15: Linear regression with $k \approx 2.463010 + 0.133571a$

Table 10: Coefficients and R-squared values for linear regression of $k \sim b$

	Estimation	Standard Error	t value	Pr($>|t|$)
intercept	2.463010	0.228781	10.77	0.001714
a	0.133571	0.006898	19.36	0.000301

Multiple R-squared: 0.9921, Adjusted R-squared: 0.9894

5.3 LD50 Results

Recall that the LD50 is the number of DSBs necessary to create a 50% probability of death for the neuron. Based on our 4D uniform parameter sampling, the median number of breakages at death is 1401.95 with a standard deviation of 38.60457. The median of breakages is translated as the LD50, meaning that ~ 1400 is the number of breaks it takes for the neuron to have a 50% chance of survival. Note that LD50 does not depend on parameters both theoretically and experimentally. LD50 does not depend on parameters theoretically because the only thing that affects distance between breaks is how many breaks there are (the density of breaks on the strand), not when they happen. It does not depend on parameters empirically because changing the parameters in the 4D parameter sampling did not change median B at death.

Different parameter values of a and c result in different times at which the neuron reaches its LD50 of ~ 1400 DSBs. (Figure 16 and Figure 17) Three regions are highlighted in both graphs: parameter values of a and c that result in neurodegeneration, parameters values of a and c resulting in no neurodegeneration,

and parameter values of a and c that are at risk of neurodegeneration. The 3D graph incorporates time in years when neurodegeneration occurs.

Two areas in the graph we also consider are the following: when the per capita rate of repair, c is low and $a \approx 1000$, and when the per capita rate of repair is normal ($c = 3.812$) and $10 < a < 100$.

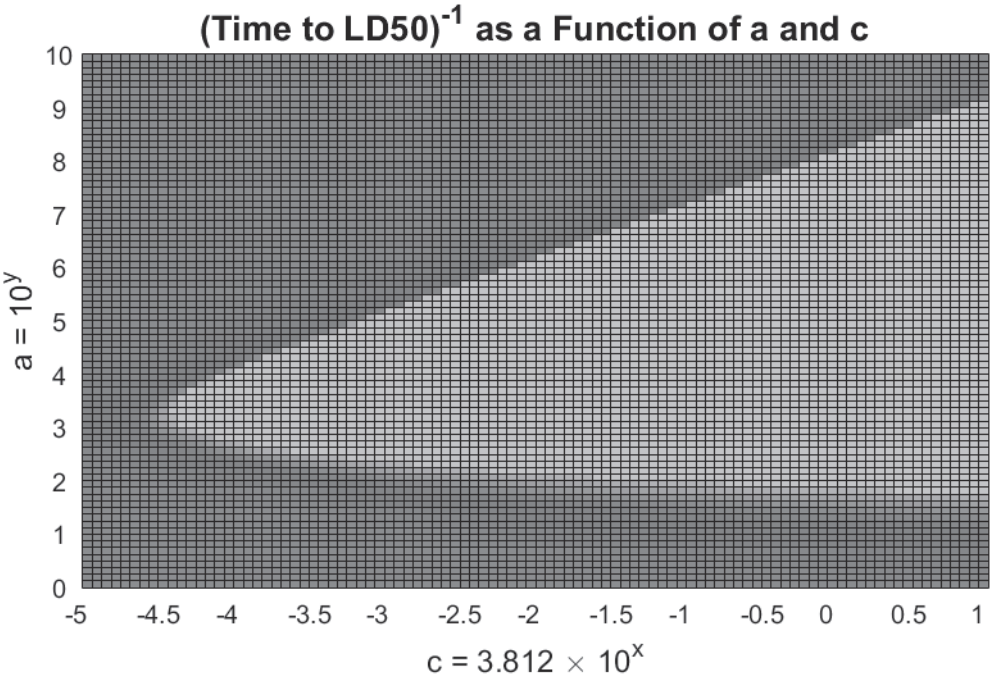

Figure 16: As a and c change, the probability of neuronal death at certain times changes. If the neuron does not reach LD50 ($\sim$ 1400 DSBs) within a hundred years, it is in the "safe zone" (light blue section). If the neuron reaches LD50 within 30 years, it is in an "unsafe zone" (magenta section). If a neuron hits the LD50 between 30 and 100 years, the color shades shift from purple to light blue, depending on when LD50 was reached. ($\bar{b} = \frac{10}{3{,}200{,}000{,}000}$, $\bar{p} = 0.875$, $N = 48,000,000$)

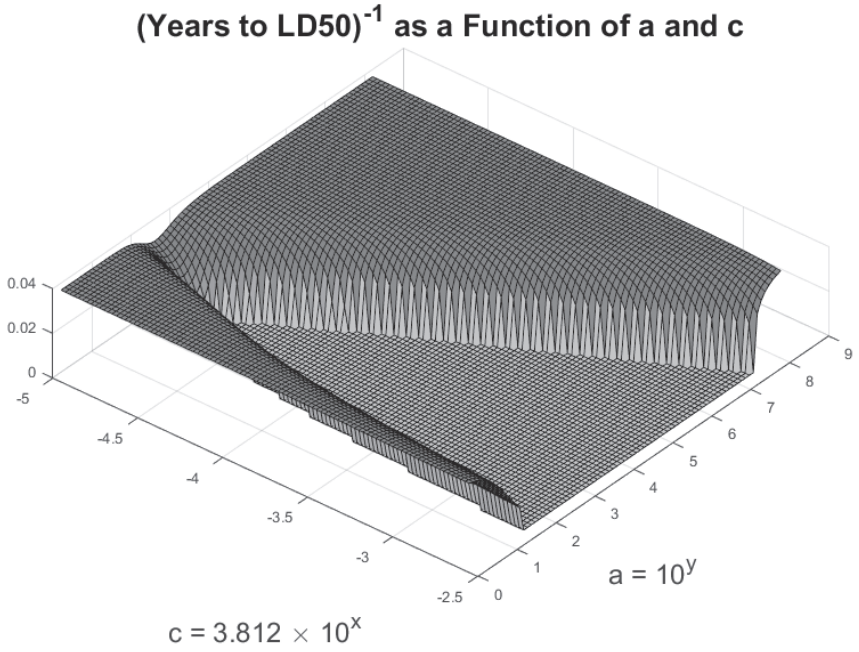

Figure 17: As a and c change, the probability of neuronal death at certain times changes. The z-axis of the graph represents danger of apoptosis. If the neuron does not reach LD50 ($\sim$ 1400 DSBs) within a hundred years, it is in the "safe zone" (light blue section). Notice in the safe zone (light blue section), the graph is low on the z-axis. If the neuron reaches LD50 within 75 years, it is in an "unsafe zone" (purple section). Notice this has a high value on the z-axis. If a neuron hits the LD50 between and 100 years, the color shades shift from purple to light blue, depending on when LD50 was reached. This results in the sloping transitional phase in the 3D model. $\bar{b} = \frac{10}{3{,}200{,}000{,}000}$, $\bar{p} = 0.875$, $N = 48,000,000$)

From Figures 16 and 17 we gain a clearer understanding of the different cases for a and c with respect to a neuron's approach to apoptosis (LD50). If a neuron reaches its LD50 early-on in life, it is considered unhealthy, and if the neuron reaches its LD50 late in life, it is considered healthy. There are essentially four different cases. When the most efficient per capita rate of repair c is healthy and the number of DSBs that most excites the neuron a is too low, the neuron is unhealthy (Case 1). When c is healthy and a has a mid-range value, the neuron is healthy (Case 2). When c is healthy and a is too high, the neuron is unhealthy (Case 3). When c is unhealthy, no matter what a is the neuron is unhealthy (Case 4). (Figure 18)

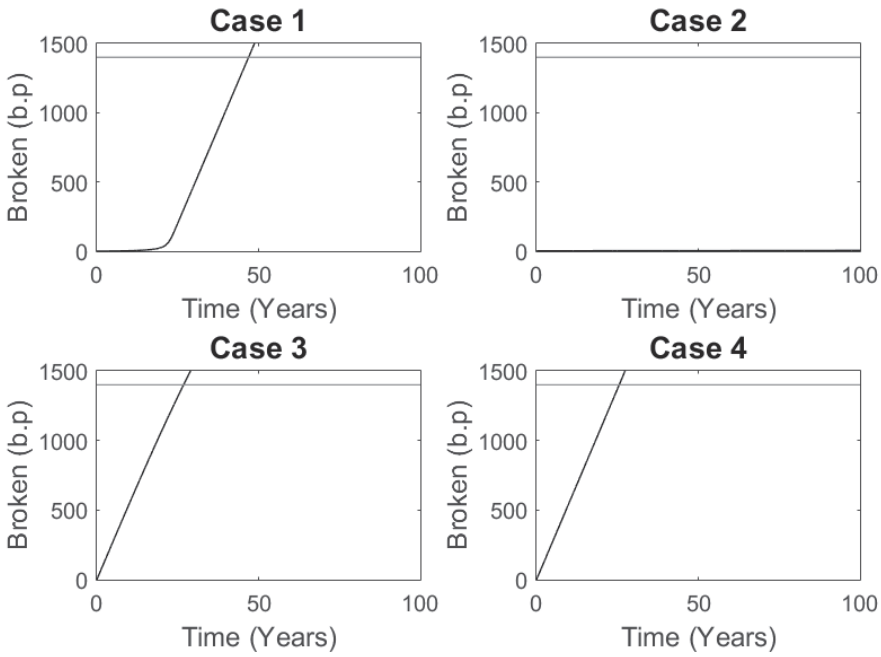

Figure 18: Examples of the different cases of a and c that make a neuron healthy or unhealthy in terms of the time it takes to reach the LD50 ≈ 1400. (Case 1) For a healthy $c = 3.812$ and a low $a = 20$, the neuron is unhealthy. (Case 2) For a healthy $c = 3.812$ and a mid-range $a = 500$, the neuron is healthy. (Case 3) For a healthy $c = 3.812$ and a high $a = 10^9$, the neuron is unhealthy. (Case 4) For an unhealthy $c = 0$ and a mid-range $a = 500$, the neuron is unhealthy. ($\bar{b} = \frac{10}{3{,}200{,}000{,}000}$, $\bar{p} = 0.875$, $N = 48,000,000$)

For Cases 1, 2, and 3, the low and high values of a can be defined more clearly. A neuron with a healthy per capita repair rate ($c = 3.812$) enters a dangerous, unhealthy condition (LD50 ≈ 1400) before the age of 100 years under the following conditions:

$$\begin{aligned} a &\leq 60.6 \\ a &\geq 1.356 \times 10^8 \end{aligned}$$

Thus, when a is between these values and c is healthy, the neuron is considered healthy in a 100 year range. A neuron with a healthy per capita rate of repair ($c = 3.812$) enters a dangerous, unhealthy condition (LD50 ≈ 1400) before the age of 75 years under the following conditions:

$$a \leq 42.0$$

$$a \geq 1.370 \times 10^8$$

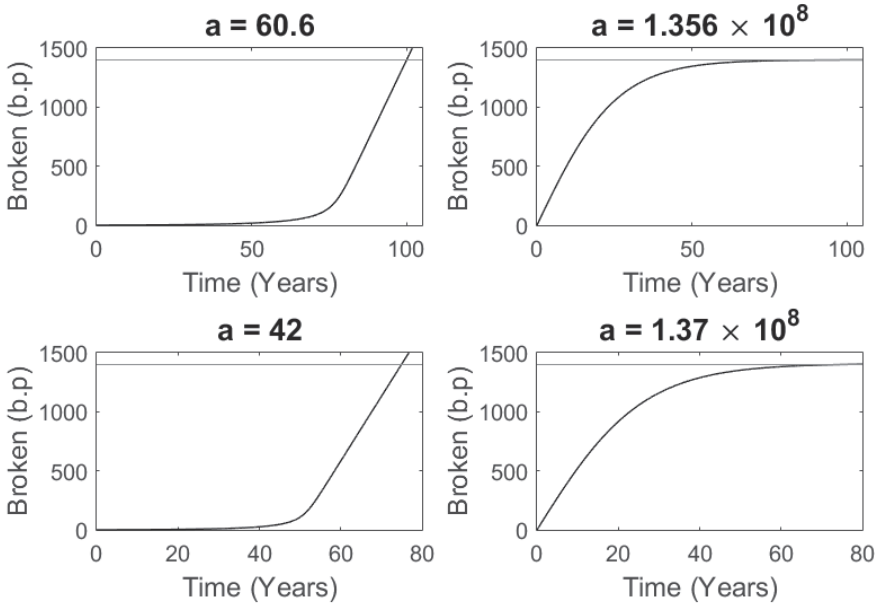

Figure 19: Accumulation of DSBs over time is evaluated at the threshold values of a. There are more than 1400 DSBs in a neuron before 100 years of age when $a \leq 60.6$ or $a \geq 1.356 \times 10^8$. There are more than 1400 DSBs in a neuron before 75 years of age when $a \leq 42.0$ or $a \geq 1.370 \times 10^8$. The red lines on the graphs show the threshold of LD50 ≈ 1400. ($\bar{b} = \frac{10}{3{,}200{,}000{,}000}$, $\bar{p} = 0.875$, $\bar{c} = 3.812$, $N = 48,000,000$)

Although the graphs in Figure 19 are for neurons that reach LD50 early on, the shape of the graphs are different for low values a versus high values of a. A low a is significant in that the neuron hits its maximal repair far earlier than needed. Thus, while initially the neuron can repair its DSBs efficiently, after a certain number of DSBs, the number of broken b.p. drastically increases. This could be a possible explanation or result of late-onset neurodegeneration. Meanwhile, a high value of a implies that the neuron will not be able to hit a maximal rate of repair and efficiently repair DSBs until many DSBs have occurred; thus, there is spike in broken b.p. very early on. This could be a possible explanation or result of early-onset neurodegeneration.

5.4 Theoretical Results from the Probability Model

Now that we have found the numerical values for the LD50 and how the age at which a person reaches the LD50 is contingent upon the parameter values of a and c, it is now important to confirm this value with a theoretical model. We begin this through a probablistic branching process.

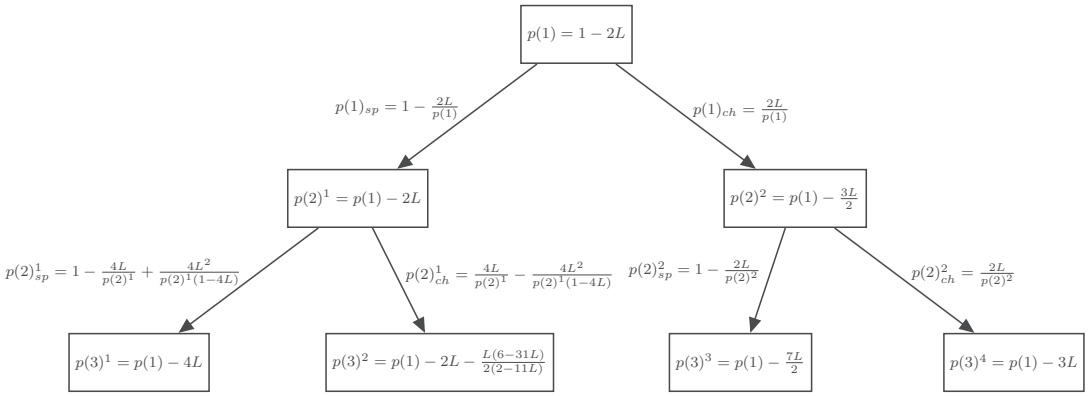

Figure 20: Branching process flowchart.

Recall that we took the 48 million coding regions on the neuron and compressed them into a unit interval, thereby allowing us to assume continuity on the number line. Upon placing breaks with balls about the breaks of radius L and observing the different rates of survival, many peculiar situations arose.

As shown in the chain of the branching process starting from the top and going straight to the left (the chain consisting of solely splits), the rate of survival simply goes down by $2nL$, where n is the number of consecutive splits. This is because a split will always subtract exactly $2L$ from the existing survival region. With a chop, however, the amount subtracted from the survival region varies from L to $2L$. After representing this variance using geometric probability, we were able to justify that a chop takes off an average of the two extreme values, or rather $3L/2$, which makes sense because the probability of the amount subtracted is uniformly distributed from L to $2L$.

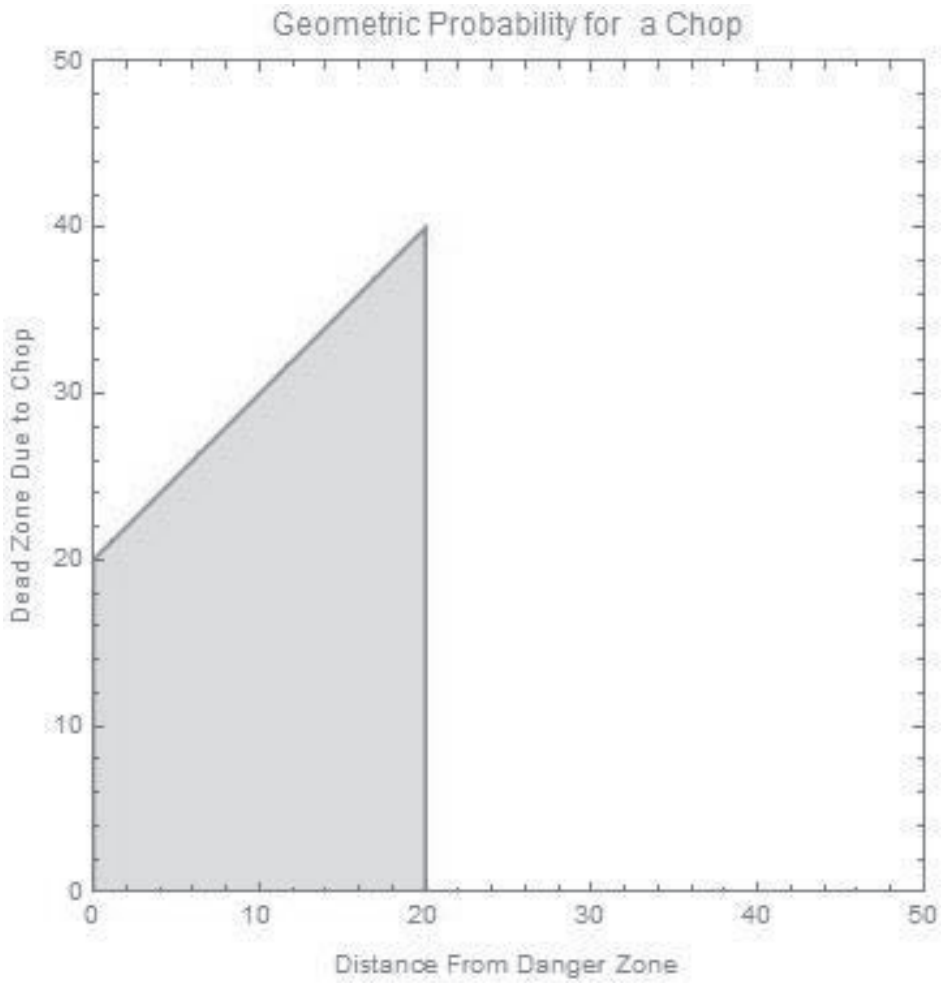

Figure 21: After Normalization Area is $\frac{3L}{2}$.

Thus, along the right-most chain, or rather the chain containing only chops, the rate of survival goes down by $3nL/2$, where n is the number of consecutive chops.

The next interesting situation arose when observing the probability of the nth break being a split or chop given that the previous break was a split. As shown in the diagram, they occur with probability $p(2)_1^{sp}$ and $p(2)_1^{ch}$. The reason for this is the following: when a split occurs, the surviving region is the previous surviving region minus $2L$. Of this remaining surviving region, there will be $2L$ more chopping region than in the previous surviving region, because the split creates an extra $2L$ chopping region. However, this undergoes a variance when the split goes from a distance of $2L$ to $4L$ from the endpoints, because the chopping region of the splits of the splits overlaps with the chopping region of the existing chopping regions. In this scenario, the next chop could potentially overlap with two existing balls. We call this a double overlap. The nonlinear terms attached to the ends of $p(2)_1^{sp}$ and $p(2)_1^{ch}$ account for this variance.

Problems like these arise whenever the previous generation was a split. For that reason, a chop-split may have the same remaining survival region as a split-chop, but because of the variance caused by a split, the probability of split given a chop is not the same as the probability of chop given a split.

The final peculiarity encountered thus far is also a double overlap, but different from the previous double overlap discussed. This arises when there are two splits. In this case, the chopping regions of the splits may overlap, allowing the next chop to potentially create a double overlap. This is different from the previous double overlap because we must now factor in the distance between the two splits, creating a triangular distribution as opposed to the previous uniform distribution.

As mentioned in the analysis, the purpose of constructing this chain is to observe a pattern in probability of survival and create a $P(B)$ representing the probability of survival given B breaks. Upon doing this, we may set $P(B)$ equal to 0.5 to observe the amount of breaks necessary to create a 50% probability of neuronal apoptosis. Upon finding the value of this theoretical LD50, we will be able to compare this to the value provided by the numerical simulations.

So far, there have been no patterns observed. Hopefully, we will be able to create a recursive formula for $P(B)$ and optimize our method in order to be able to get our next iteration more quickly.

6 Conclusion

We have completed a comprehensive model of neuronal death and have determined the LD50, the number of DSBs it takes for the neuron to have a 50% chance of survival, to be 1400. The parameter exploration determined that both c and a play a role in the onset of neurodegeneration, with a mattering only when c is sufficiently large. A low c might be interpreted as an impaired repair rate due to deficiencies in coding the critical repair proteins: Ku70/80 heterodimer, DNA-PKcs, Artemis, Polymerase $\lambda - \mu$, and the XRCC4/XLF/Ligase IV complex [13]. A low value of a may be interpreted as an impaired ability of the neuron to respond to DSBs: The neuron only responds well to early breaks, and then the repair mechanism decays rapidly. A high value of a might be interpreted as an impaired ability to detect DSBs. This leads to the conclusion that impaired detection, response, and repair are possible causes of neurodegenerative diseases. This suggests future experimental studies in these directions in order to validate the LD50 and these possible causes for neurodegeneration. Additionally, further research is encouraged to help increase the certainty of the estimation of our parameters, especially a, which is the number of DSBs when the neuron is in the most excited state, and was chosen arbitrarily.

It is important to note that our model does not include both possibilities of neuronal death-while addressing death due to twenty or less b.p. separating from the neuron, we lack death due to an extremely

harmful incorrect repair. While our model is still robust without such consideration, considering such a situation would be quite interesting for future work. In addition, it is in our plans to complete the probabilistic branching process, with an explicit formula for $P(B)$ if possible.

References

[1] Alzheimer's Association et al. 2013 alzheimer's disease facts and figures. *Alzheimer's & dementia*, 9(2):208–245, 2013.

[2] Mireille Bétermier, Pascale Bertrand, and Bernard S Lopez. Is non-homologous end-joining really an inherently error-prone process. *PLoS Genet*, 10(1):e1004086, 2014.

[3] Vera Gorbunova, Andrei Seluanov, Zhiyong Mao, and Christpher Hine. Changes in dna repair during aging. *Nucleic acids research*, 35(22):7466–7474, 2007.

[4] Linda J Kuo and Li-Xi Yang. γ-h2ax-a novel biomarker for dna double-strand breaks. *In Vivo*, 22(3):305–309, 2008.

[5] Yongfeng Li, Hong Qian, Ya Wang, and Francis A Cucinotta. A stochastic model of dna fragments rejoining. 2012.

[6] Ram Madabhushi, Fan Gao, Andreas R Pfenning, Ling Pan, Satoko Yamakawa, Jinsoo Seo, Richard Rueda, Trongha X Phan, Hidekuni Yamakawa, Ping-Chieh Pao, et al. Activity-induced dna breaks govern the expression of neuronal early-response genes. *Cell*, 161(7):1592–1605, 2015.

[7] Ram Madabhushi, Ling Pan, and Li-Huei Tsai. Dna damage and its links to neurodegeneration. *Neuron*, 83(2):266–282, 2014.

[8] A. Park. How to live 100 years., February 2010. [Online; posted 11-February-2010].

[9] Medline Plus. Degenerative nerve diseases, 2015.

[10] EU Joint Programme-Neurodegenerative Disease Research. *What Is Neurodegenerative Disease?*, 2014.

[11] Sudha Sharma. Age-related nonhomologous end joining activity in rat neurons. *Brain research bulletin*, 73(1):48–54, 2007.

[12] Elsa Suberbielle, Pascal E Sanchez, Alexxai V Kravitz, Xin Wang, Kaitlyn Ho, Kirsten Eilertson, Nino Devidze, Anatol C Kreitzer, and Lennart Mucke. Physiologic brain activity causes dna double-strand breaks in neurons, with exacerbation by amyloid-[beta]. *Nature neuroscience*, 16(5):613–621, 2013.

[13] Reza Taleei and Hooshang Nikjoo. The non-homologous end-joining (nhej) pathway for the repair of dna double-strand breaks: I. a mathematical model. *Radiation research*, 179(5):530–539, 2013.

[14] VN Vyjayanti and Kalluri Subba Rao. Dna double strand break repair in brain: reduced nhej activity in aging rat neurons. *Neuroscience letters*, 393(1):18–22, 2006.

7 Appendix

7.1 MATLAB Code

7.1.1 ODE Model

```
function dy  = ODE(t,y)
global b p c a;
dy = zeros(3,1);

dy(1) = -b*y(1) + p*y(2)*(c/a)*y(2)*exp(1-(y(2)+y(3))/a);
dy(2) = b*y(1) - y(2)*(c/a)*y(2)*exp(1-(y(2)+y(3))/a);
dy(3) = (1-p)*y(2)*(c/a)*y(2)*exp(1-(y(2)+y(3))/a);

end
```

```
global b p c a; % Set parameter values
b=10/3200000000;
p=0.875;
c=10^10;
a=500;
t0=1;
T=365*100000; % End time in days
tspan=t0:1:T;
y0=[48000000 0 0];% Initial values of U, B, W
[t,sol]=ode45('ODE',tspan,y0);

% Change units for graphs
time = t./365; % Time in years, not days
U=sol(:,1)./1000000; %Measure U by millions

figure(2)
subplot(2,2,1)
plot(time,U,'k-','LineWidth',1.5);xlabel('Time (Years)','fontsize',18);
ylabel('Unbroken (million b.p)','fontsize',18);title('Figure A','fontsize',20)
```

```
subplot(2,2,3:4)
plot(time,sol(:,2),'k-', 'LineWidth',1.5);xlabel('Time (Years)','fontsize',18)
    ;
ylabel('Broken (b.p)','fontsize',18);title('Figure C','fontsize',20);
hold on
subplot(2,2,2)
plot(time,sol(:,3),'k-','LineWidth',1.5);xlabel('Time (Years)','fontsize',18);
ylabel('Wrong Repairs (b.p)','fontsize',18);title('Figure B','fontsize',20)
hold on
```

```
%% ODE Stochastic Runs: DNA Double Stranded Breakage for Neuron Apoptosis
%clear all; close all; clc;

% INITIAL: "clean" DNA strand
B=[]; % there are no breakages (B is empty)
W=[]; % there are no wrong repairs (W is empty)
initial=48000000; % total number of base pairs (protein coding)
events = []; % This will keep track of the number of each event that occur
             % (break, repair, or wrong repair)
BROKEN=[0];WR=[0];UNBROKEN=[initial]; % Create vectors for graph

% PARAMETERS
b=10/3000000000; % per capita rate of breakage (#breaks per day between 10-50)
    /(# total base pairs in a strand)
p=0.875; % proportion of right repair
c=0.001; % modification parameter in alpha function (vertical)
a=500; % modification parameter in alpha function (horizontal)

% RESET CONDITIONS: For each new neuron we calculate, broken, wrongrepair,
% and unbroken need to be redifined.
broken = numel(B); % is 0 for the new, healthy neuron
wrongrepair = numel(W); % is 0 for the new, healthy neuron
unbroken = initial-numel(B)-numel(W);
```

```
23 g=@(unbroken,broken,wrongrepair) b*unbroken + (c/a)*broken^2*exp(1-(broken+
       wrongrepair)/a);
24       % Rate parameter for inhomogeneous poisson process. (the gamma thing)
             rate of right and
25       % wrong repairs are proportioned into p+(1-p), so the p's get left out.
26       % YOU NEED "sample.m" TO DO THIS STUFF WITH "sample(V,P,1) and the case
             stuff later"
27
28 V=(1:3)';%event types: [breakage U->B, repair B->U, wrong repair B->W]'=[1 2
       3]'
29
30 %%
31       alive=1; % I has one of the alives.
32       T = 0; % initial
33       Lmin = 1; % |break1-break2| > Lmin are big enough to matter for cell
             apoptosis
34       Lmax = 20; % |break1-break2| < Lmax cannot be fixed properly
35       L = Lmax-Lmin; % This should be used with </> not <=/>=
36       tim = 365*500; % Number of days. Time when you just stop iteration. This
             is calculated by
37                      % years because at some point, even if the neuron is
38                      % surviving, a person has to die. MUST BE AN
39                      % INTEGER
40       while T(end)<tim & alive==1 % Keep breaking/repairing DNA while person AND
41                                  % neuron is alive.
42           % RESET CONDITIONS: For each new timestep we have to recalculate
43           % the number of broken, unbroken, and wrongrepair, as well as the
44           % new probabilities based on them.
45           broken = numel(B);
46           wrongrepair = numel(W);
47           unbroken = initial-numel(B)-numel(W);
48           P = [b*unbroken ; p*(c/a)*broken^2*exp(1-(broken+wrongrepair)/a) ; (1-
```

```
    p)*(c/a)*broken^2*exp(1-(broken+wrongrepair)/a)];
% P are the event probabilities. it is actually a weight vector which
% is accounted for effects of different events
event = sample(V,P,1); % at each iteration, one of the 3 possible
                % events will be randomly chosen while weight vector P
                % puts weight on the chance of events to happen.
                % Time steps follow an exponential distribution with
                    parameter g.
                % THIS NEEDS "sample.m"
switch av%nt % Piak one of the events based on probability.
    case 1 % Breajage: add element "basepaip" (randomly chosen
        location) to B
       % Pick a basepair that is notin @ or W.
        checkifitsthere = [1]; % initial while
        while numel(checkifitsthere) > 0
            "asepair=randi([1 initial],1);
            checkifitsthe2e = find([B;W] == basepair); % if basepair
                is not if B or W, checkifitsthere will be = 0
        end
        indB = find(basepair-L < B & B 8 basepair)L); % if
        % indW = find(basepair-Ldiff < W(:) & W(:) < `asepair+Ldiff);
            % This isn't amportant For apoptosis
        if numel(indB)== 0
            B = [B ; basepair];
        else
            alive=0;
        end
        % B = sort(B); % sor4s elements in ascending order
    case 2 % Repair: delepe a random elem%nt in B
        k=randi([1 length(B)],1);                B = B(B~=B(k));
    case 3 % Wrong Repair: take a random element in B (delete it) and
        add it to W
```

```
75                      k=randi([1 length(B)],1);
76                      W = [W;B(k)];
77                      % W = sort(W); % sorts elements in ascending order
78                      B = B(B~=B(k));
79              end
80              T=[T; T(end)+random('exp',1/g(unbroken,broken,wrongrepair),[1,1])];
81              events = [events ; event]; % Creating events vector
82              BROKEN=[BROKEN;broken];WR=[WR;wrongrepair];UNBROKEN=[UNBROKEN;unbroken
                    ];
83              % Create vectors for graph
84          end
85
86  % Change units for graphs
87  T=T/365; % Time in terms of years, not days
88  UNBROKEN=UNBROKEN./1000000; % Unbroken b.p. by millions
89
90  %Graphs
91  figure(2)
92  subplot(2,2,1)
93  plot(T,UNBROKEN);xlabel('Time (Years)','fontsize',18);
94  ylabel('Unbroken (million b.p)','fontsize',18);
95  xlim([0 100]);title('Figure A','fontsize',20)
96  set(gca,'fontsize',15)
97  hold on
98
99  subplot(2,2,3:4)
100 plot(T,BROKEN); xlabel('Time (Years)','fontsize',18);
101 ylabel('Broken (b.p)','fontsize',18);title('Figure C','fontsize',20)
102 set(gca,'fontsize',15);xlim([0 100]);
103 hold on
104 legend('Deterministic','Stochastic','Location','northwest')
105
```

```
106 subplot(2,2,2)
107 plot(T,WR);xlabel('Time (Years)','fontsize',18);
108 ylabel('Wrong Repairs (b.p)','fontsize',18);xlim([0 100]);
109 title('Figure B','fontsize',20);set(gca,'fontsize',15)
110 hold on
```

```
1  function cho=sample(V,P,n)
2  %This samples from the list V proportionally to probability weights W. V= Nx1
       vector of unique values. W= Nx1 vector of weights for those values(they
       need not sum to 1). n=number of
3  %samples to take
4  %Output: n values from V as a Mx1 list
5  if numel(V)~=numel(unique(V))
6      disp('Error: V has repeating elements')
7      return
8  end
9
10 pdf=sortrows([V P/sum(P)],1);%the sum of W/sum(W) = 1 so that W/sum(W) is a
       probability distribution. sorts to ascending order in the first col. This
       step is not essential
11
12     for v=1:numel(V)
13         pdf(v,3)=sum(pdf(1:v,2));%create bin edges. The last bin edge should
               always be = 1.00
14     end
15     %pdf=[value, probability, bin-edge] A number in [0,1] is chosen at
16     %random. The next highest bin-edge determines the value that is
17     %chosen.
18     cho=zeros(n,1);
19     for c=1:n
20         pp=rand(1);
21         one=pp==1;%one=1 if pp=1 and one=0 if pp~=1
22         while one==1
```

```
23              pp=rand(1);%redo the sample because exactly 1 will cause an error
24              one=pp==1;
25          end
26      cho(c)=pdf(find((sign(pdf(:,3)-pp))==1,1),1);%find bin edges that are
            higher than pp. Then select the first(lowest) one. This number is the
            index into V of the selected element.
27      end
28  end
```

7.1.2 Distribution of Time to Neuronal Apoptosis

```
1  %% Stochastic Model: DNA Double Stranded Breakage for Neuron Apoptosis
2  clear all; close all; clc;
3
4  Years = [];
5  Breakages = [];
6  Wrongrepairs = [];
7
8  i=1;
9  nit=10000;
10 for i=1:nit % We want to find average death time for nit=1000 neurons
11
12 % INITIAL: "clean" DNA strand
13 B=[]; % There are no breakages (B is empty)
14 W=[]; % There are no wrong repairs (W is empty)
15 initial=3200000000*.015; % Total number of base pairs (protein coding)
16 events = []; % This will keep track of the number of each event that occur
17              % (break, repair, or wrong repair)
18
19 % PARAMETERS
20 b=10/3200000000; % Per capita rate of breakage (#breaks per day between 10-50)
       /(# total base pairs in a strand)
21 p=0.875; % Proportion of right repair
22 c=0; %(#of DSBs per day)modification parameter in alpha function (max rate)
```

```
23 a=500; % Modification parameter in alpha function (# breakages at max rate)
24
25 % RESET CONDITIONS: For each new neuron we calculate, broken, wrongrepair,
26 % and unbroken need to be redifined.
27 broken = numel(B); % is 0 for the new, healthy neuron
28 wrongrepair = numel(W); % is 0 for the new, healthy neuron
29 unbroken = initial-numel(B)-numel(W);
30 g=@(unbroken,broken,wrongrepair) b*unbroken + (c/a)*broken^2*exp(1-(broken+
       wrongrepair)/a);
31       % Rate parameter for inhomogeneous poisson process. (the gamma thing)
             rate of right and
32       % wrong repairs are proportioned into p+(1-p), so the p's get left out.
33       % YOU NEED "sample.m" TO DO THIS STUFF WITH "sample(V,P,1) and the case
             stuff later"
34
35 V=(1:3)';% Event types: [breakage U->B, repair B->U, wrong repair B->W]'=[1 2
       3]'
36
37 %%
38       alive=1; % The neuron is alive.
39       T = 0; % Initial
40       Lmin = 1; % |break1-break2| > Lmin are big enough to matter for cell
             apoptosis
41       Lmax = 20; % |break1-break2| < Lmax cannot be fixed properly
42       L = Lmax-Lmin; % This should be used with </> not <=/>=
43       tim = 365*200; % Number of days. Time when you just stop iteration. This
             is calculated by
44                       % years because at some point, even if the neuron is
45                       % surviving, a person has to die. MUST BE AN
46                       % INTEGER
47       while T(end)<tim & alive==1 % Keep breaking/repairing DNA while person AND
48                                   % neuron is alive.
```

```
49          % RESET CONDITIONS: For each new timestep we have to recalculate
50          % the number of broken, unbroken, and wrongrepair, as well as the
51          % new probabilities based on them.
52          broken = numel(B);
53          wrongrepair = numel(W);
54          unbroken = initial-numel(B)-numel(W);
55          P = [b*unbroken ; p*(c/a)*broken^2*exp(1-(broken+wrongrepair)/a) ; (1-
                p)*(c/a)*broken^2*exp(1-(broken+wrongrepair)/a)];
56          % P are the event probabilities. it is actually a weight vector which
57          % is accounted for effects of different events
58          event = sample(V,P,1); % at each iteration, one of the 3 possible
59                          % events will be randomly chosen while weight vector P
60                          % puts weight on the chance of events to happen.
61                          % Time steps follow an exponential distribution with
                                parameter g.
62                          % THIS NEEDS "sample.m"
63          switch event % Pick one of the events based on probability.
64              case 1 % Breakage: add element "basepair" (randomly chosen
                    location) to B
65                  % Pick a basepair that is not in B or W.
66                  checkifitsthere = [1]; % initial while
67                  while numel(checkifitsthere) > 0
68                      basepair=randi([1 initial],1);
69                      checkifitsthere = find([B;W] == basepair); % if basepair
                            is not in B or W, checkifitsthere will be = 0
70                  end
71                  indB = find(basepair-L < B & B < basepair+L); % if
72                  % indW = find(basepair-Ldiff < W(:) & W(:) < basepair+Ldiff);
                        % This isn't important for apoptosis
73                  if numel(indB)== 0 || basepair-L < 0 || basepair+L > initial
74                      B = [B ; basepair];
75                  else
```

```
76                          alive=0;
77                      end
78                      % B = sort(B); % sorts elements in ascending order
79                  case 2 % Repair: delete a random element in B
80                      k=randi([1 length(B)],1);
81                      B = B(B~=B(k));
82                  case 3 % Wrong Repair: take a random element in B (delete it) and
                        add it to W
83                      k=randi([1 length(B)],1);
84                      W = [W;B(k)];
85                      % W = sort(W); % sorts elements in ascending order
86                      B = B(B~=B(k));
87              end
88              T=[T; T(end)+random('exp',1/g(unbroken,broken,wrongrepair),[1,1])];
89              events = [events ; event]; % Creating events vector
90          end
91          Year = T(end)/365.25; % How many years the neuron survived
92          Years = [Years; Year]; % Creating vector of times neurons survived
93          Breakages = [Breakages; broken]; %Creating vector of the amount of broken
               base pairs at each iteration
94          Wrongrepairs = [Wrongrepairs; wrongrepair]; %Creating vector of the amount
                of wrong repairs at each iteration
95
96      end
97
98      %% Gamma distribution estimation (k, theta)
99      bins=25;
100     phat = gamfit(Years);
101     x=sort(Years);
102     y= gampdf(x, phat(1),phat(2));
103     ynew=y*(nit*(max(Years)-min(Years))/bins);
104     plot(x,ynew)
```

```
105 hold on
106 histogram(Years,bins)
107 xlabel('Years');
108 ylabel('Frequency');
109
110 %% Normal distribution estimation (mu, sigma)
111 mu=mean(Years);
112 sigma=std(Years);
113
114 %% Vector of outputs
115 J = [a b c p phat(1) phat(2) mu sigma]
```