APPROVAL SHEET

Title of Thesis: Use of Systems Engineering Approaches To Examine Stability & Sensitivity Of Alzheimer's Associated Pathways

Name of Candidate: Neil Tushar Agarwal
Master of Science, 2016

Thesis and Abstract Approved: Theresa A. Good

Theresa A. Good
Professor
Department of Chemical, Biochemical & Environmental Engineering

Date Approved: 4/29/16

ABSTRACT

Title of Document: USE OF SYSTEMS ENGINEERING APPROACHES TO EXAMINE STABILITY & SENSITIVITY OF ALZHEIMER'S ASSOCIATED PATHWAYS

Neil Tushar Agarwal, MS, 2016

Directed By: Professor, Theresa Good

Department of Chemical, Biochemical & Environmental Engineering

Alzheimer's disease (AD) is a progressive neurodegenerative disorder that affects the elderly population. There are two histopathological hallmarks of AD, the formation of neurofibrillary tangles, and amyloid plaques. The primary protein component of senile plaques is beta amyloid (Aβ), a 39 to 43 amino acid long peptide, which investigators believe plays a causative role in AD. At a molecular level, it appears that Aβ impacts complex signaling networks that contain a substantial degree of signal integration. To date, most investigators have examined the influence of Aβ by investigating one signaling pathway at a time.

In this work, a systems engineering approach is taken to examine how much Aβ influences the stability and sensitivity to change of biological parameters, and discuss the possible biological interpretations in a complex reaction network.

The simplified model did not capture expected trends. Viability and death signals were insensitive to the amount of ECM and Aβ, although the output was sensitive to the rate constant associated with matrix-integrin interaction. Our more complex model did capture expected salient trends with respect to the viability signal. Furthermore, the model displayed asymptotic stability at high and low viability signal in the absence and presence of Aβ respectively. Sensitivity to parameter interactions associated with Aβ were observed, that could be related to experimental toxicity attenuation data.

The work demonstrates the utility of such tools in analyzing reaction networks. The tools can relate model parameters with experimental findings, with improved predictions, that help identify therapeutic avenues for altering neurotoxicity associated with Aβ.

USE OF SYSTEMS ENGINEERING APPROACHES TO EXAMINE STABILITY & SENSITIVITY OF ALZHEIMER'S ASSOCIATED PATHWAYS

By

Neil Tushar Agarwal

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Master of Science
2016

Dedication

To my loving parents and grandmother,
Arun, Manorama and Parvati
for their unwavering love, support and for being amazing cooks.

To my dear friend and comrade in arms,
Raymond "Fuego" Fioravante

To my mentor, colleague, and friend,
Theresa A. Good,
for playing a pivotal role in making me the man I am today.

Acknowledgements

First, I would like to thank my mentor Dr. Theresa Good, who has supported and advised me throughout the course of my MS. Since 2006, my relationship with her has grown from mentor, to colleague, to friend in the course of 10 years. She has given me latitude to pursue my scientific research, and has always supported my decisions in academic and other pursuits as well. I would also like to thank my department faculty, research group members and colleagues at UMBC, for making my tenure at UMBC a worthwhile and enrichening experience, without whom completing graduate school would not have been possible.

Specifically, I want to thank my oral qualifying exam and defense committee comprising of Dr. Jennie Leach, Dr. Mariajose Castellanos, Dr. Christopher Hennigan, Dr. Lee Blaney, Dr. Timothy Topoleski and Dr. Douglas Frey. Their critical analysis of my research work, inputs and regular feedback have greatly enhanced my learning process in a positive manner, and helped me grow as a researcher. I would like to thank Douglas Frey, who as the Graduate program director, made me feel as a part of the UMBC Chemical Engineering community since I joined this program in 2013. He played an indispensable role into facilitating my transition from a non-degree student to a thesis candidate.

Many thanks also to Denise Kedzierski, for processing my documentation and provided important administrative support which ensured my stay at UMBC was smooth and comfortable, especially near the end of my defense.

I want to thank my labmates Dr Natasha Wilson and Dr. Arundathi Venkatasubramanian who trained me in biological background, experimental procedures, and provided me wonderful company when working late nights and long days at school. I also wish to acknowledge the help and contribution of undergraduate students who had a major part in designing the sensitivity model, namely Raymond Fioravante.

I acknowledge my fellow graduate students at UMBC, Cindy Chelius, Ke He, Sheniqua Brown, Yang Liu, Aaron Gibson, Heather Harshbarger, Michael Zhang, Karthik Boppidi, Liliane Ribeiro, Michael Battaglia, and Adil Zuber for providing me friendship.

My dear friends Raymond Fioravante, John Alex Packard and Kristian Bolino for helping me through difficult times, and being there for me to celebrate the good times.

I would like to thank my parents Arun and Manorama, and my grandmother, Parvati, for their utmost support and encouragement while I worked towards my MS.

Last but not the least I would like to thank all my family and friends whom I have not mentioned by name here but who have contributed towards my work. These years are an integral and joyous part of my life.

Table of Contents

List of Tables

List of Figures

Chapter 1: Biological Background

Motivation Behind Work and Summary

Alzheimer's disease (AD) is a neurodegenerative disease that affects millions across the world and no effective treatment has yet been identified for the disease (Shankar et al., 2007). However, we do not fully understand the mechanism behind the disease. In our work, several proposed biochemical mechanisms, with increasing complexity, due to incorporating pertinent metabolites, were proposed and applied tools from systems engineering to study the molecular mechanism of cell signaling associated this disease. The subject of this research was the beta-amyloid (Aβ) peptide in its aggregated form, which is hypothesized to induce a chain of events that lead to neurodegeneration in AD.

We first investigated how Aβ interacted with cells by proposing several biochemical mechanisms that included signaling molecules that were found to be relevant by previous students in our laboratory. Key signaling molecules included G-Protein Coupled Receptors (GPCRs), Src family kinases, Integrins, and extracellular matrix, which were Different students in that laboratory had found each of these to be associated with the toxicity of aggregated Aβ (Rymer et al., 2001; Venkatasubramanian et al., 2014). Once the biochemical mechanisms were developed, systems biology techniques were utilized to examine stability behavior of mathematical models. Finally, a sensitivity analysis was conducted on various parameters such that we could discriminate model performance among various proposed mechanisms, and infer the effect of changes on key model outputs such as

viability and death signals. This work contributes new systems level tools to the toolkit that may guide the design of new experiments or assist in the generation of new hypotheses that will contribute to our understanding of the role of Ab in cell signaling associated with neurotoxicity in Alzheimer's disease. Similar tools could be used to explore the behavior of many non-linear dynamic systems that have relevance in human health, the environment, and many other areas of science that impact the quality of life and the health of this planet.

Introduction to Alzheimer's disease (AD) and Aß

Alzheimer's disease (AD) is a form of dementia that affects memory, thinking and behavior with symptoms that gradually worsens over time (Klafki et al., 2006). AD is the most common form of late-life dementia, affecting 4 million Americans and over 30 million individuals worldwide (Ferri et al., 2005). Although the molecular precipitants of AD are unknown in most patients, extensive research indicates that the Amyloid-β (Aβ) protein plays a major role in pathogenesis; thus we believe there is an opportunity to use systems engineering approaches to explore the impact of Aβ on cell signaling that precedes the development of disease (Shankar et al., 2007).

Aβ is generated from the amyloid precursor protein (APP) by enzymatic digestion involving β and γ-secretases (Dickson, 1997; Hardy et al., 2002). Aβ molecules can aggregate to form flexible soluble oligomers which may exist in several forms. Misfolded oligomers can also propagate misfolding into other Aβ oligomers, leading to a cascading reaction; resulting in amyloid plaques that are toxic

to nerve cells (Nussbaum et al., 2013). The other protein implicated in Alzheimer's disease is tau, a hyper-phosporylated version of which makes up neurofibrillary tangles found in diseased patients' brains. Tau also forms misfolded oligomers, and there is some evidence that misfolded Aβ can induce tau to misfold (Nussbaum et al., 2013). Although the specific molecular initiators of AD are largely unknown, biochemical studies indicate that the severity of cognitive impairment from AD correlates more strongly with the cortical levels of soluble amyloid protein (Li et al., 2010). However, the mechanism by which the peptide causes the neurodegeneration and apoptosis of cells observed in the disease is unclear and methods to prevent its toxicity are yet to be confirmed (Li et al., 2010).

Role of Aβ in AD: learning and memory, and the potential role for abnormal phosphorylation

Learning and memory are the first set of processes that AD affects (Wang et al., 2002). Additionally, both mental processes are linked directly to long term potentiation (LTP), which is a candidate for synaptic mechanism, a strengthening of synapses based on recent patterns of activity. These are patterns of synaptic activity that produce a long-lasting increase in signal transmission between two neurons (Cooke et al., 2006). Its induction and expression is regulated by N-Methyl-D-aspartate receptor (NMDA) and α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA) receptors respectively (Jones et al., 1991). Additionally, phosphorylation has been shown to heavily influence LTP and neuronal function through the regulation of AMPA and NMDA receptors (Kalia et al., 2004). Phosphorylation affects a very large number of intracellular proteins, and is arguably

the most widely studied post-translational modification (Kalia et al., 2004). More specifically, it is substantiated by previous work which has shown that protein tyrosine kinase (PTK) inhibitors can inhibit the induction of LTP, specifically, tyrosine kinase Src was shown to regulate NMDA activity, which affects LTP (Kalia et al., 2004). Since phosphorylation directly affects LTP, our interests lie in creating a mathematical model to better understand how Aβ adversely affects phosphorylation, which we infer then alters the function of receptors like AMPA and NMDA. The short term impact of abnormal phosphorylation could be alterations in LTP, impacting learning, or alterations in NMDA and AMPA activity, also impacting learning, but long term (or acutely), alterations in NMDA activity or AMPA activity could lead to loss of calcium homeostasis and neurotoxicity.

Aβ, tyrosine kinases, G-Proteins, and signal integration

A number of studies have shown the potential integration of signaling in pathways containing Src, other tyrosine kinases, G-Proteins, and Aβ (Ittner et al., 2010). Studies have shown that Aβ may lead to activation of a tyrosine kinase in the Src family, which results in its phosphorylation to form phosphorylated Src (pSrc), that can then phosphorylate the NR2B subunit of the NMDA receptor (NMDAR) (Ittner et al., 2010). Specifically, immune-precipitation of Fyn, a member of the Src family, was found to have increased association with focal adhesion kinase (FAK) in its phosphorylated (activated) form following treatment with Aβ (Zhang et al., 1996). In addition, tyrosine phosphorylation of tau was also found in neurons where there was an increase in the tyrosine phosphorylation and activation of FAK; suggesting that

FAK/Fyn/PI3-kinase association is upregulated in AD-afflicted neurons (Zhang et al., 1996). However, little is known about the manner in which Aβ binds to cells and mediates Src family kinase phosphorylation pathways. Nevertheless, it is a reasonable assumption that Aβ, by some mechanism, activates a Src family tyrosine kinase, leading to subsequent alterations in NMDAR function, thus affecting the function of LTP, learning and memory.

An interaction between Aβ and integrins could provide the link between signaling pathways that include Src, especially Fyn (Zhang et al., 1996). Integrins are a group of transmembrane cell adhesion proteins that play an important role in mediating cell adhesion to the extracellular matrix (ECM) (Hynes et al., 2002). Integrins transmit signals bi-directionally through initiate 'inside-out signaling'; inducing the binding of talin and kindlin to the cytoplasmic domains of integrin b subunits, which activates ligand binding of integrins (Hynes et al., 2002). Conversely, the interaction between integrins and their various ligands induces 'outside-in' signals across the membrane, inducing cell spreading, retraction, migration, proliferation, and survival (Shen et al., 2012). The Aβ peptide has been shown to bind to integrins through its RHD sequence, providing another avenue for Aβ induced signaling (Sabo et al., 1999). This has been shown from previous work that Aβ interacts with or near the α6 subunit of integrin receptor, in model cell lines, neurons, and glia. Others have shown how this interaction may be important in signaling associated with NMDAR and could have implications for learning and memory in AD (Sabo et al., 1999).

Src family kinases play roles beyond the phosphorylation of NMDAR. Fyn, a member of the Src family, can activate other kinases like the Glycogen synthase kinase-3 (GSK3) and Cyclin-dependent kinase 5 (cdk5), both kinases known to phosphorylate tau at disease-associated epitopes (Zhang et al., 1996). Hence Fyn, has been connected to both Aβ and tau, and forms the common thread between two major pathological hallmarks of AD.

G protein-coupled receptors (GPCRs) are GTPases (guanosine triphosphatases) that cycle between a GDP-bound form and a GTP-bound form, which are involved in numerous key neurotransmitter systems in the brain that are disrupted in AD (Thathiah et al., 2011). They are some of the most widely reported membrane receptors in the literature because of their widespread involvement in signaling pathways associated with health and disease. The GTP-bound G protein is an active form that interacts with downstream effectors and transmits signals, during which the bound GTP is often hydrolyzed to GDP, and then recycles into the inactive GDP-bound form. The activity of G proteins is regulated mainly through three classes of regulatory proteins: GTPase activating proteins (GAPs), guanine nucleotide-exchange factors (GEFs), and guanine nucleotide-dissociation inhibitors (GDIs) (Shen et al., 2012).

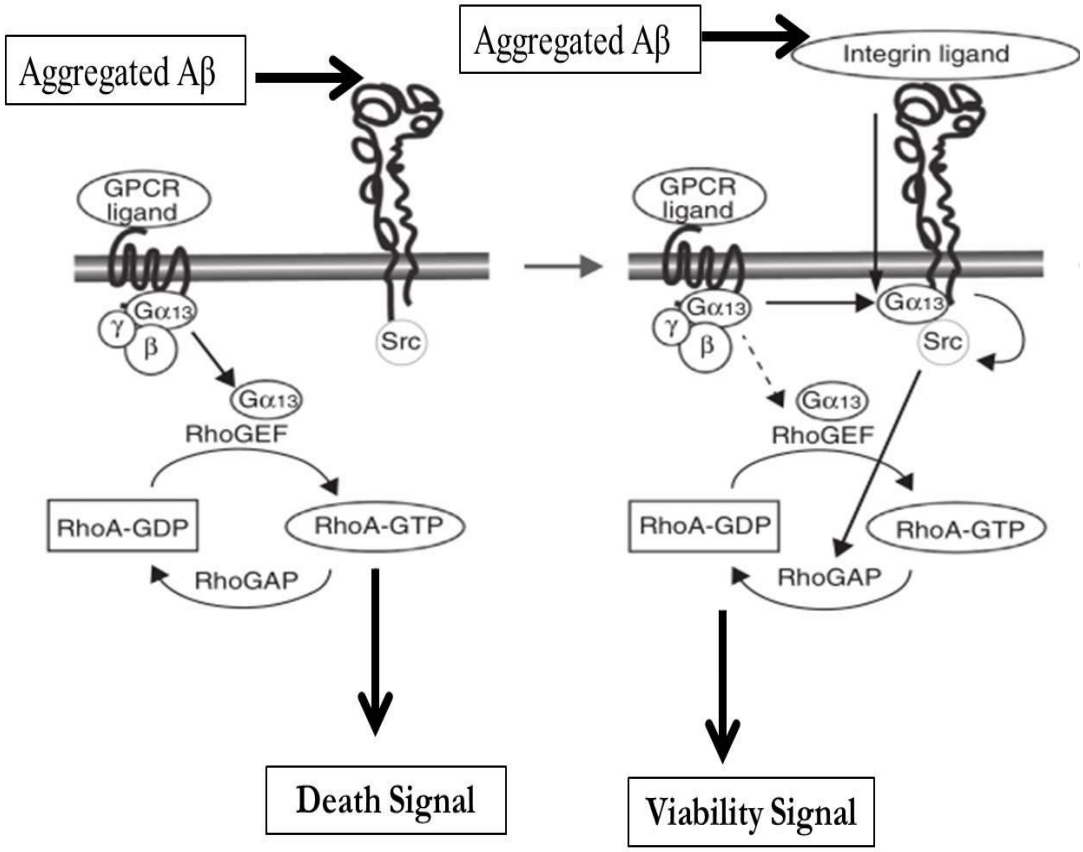
GPCRs directly influence the amyloid cascade through modulation of the α, β and γ-secretases, proteolysis of the amyloid precursor protein (APP), and regulation of Aβ degradation. Several studies have presented compelling evidence implicating GPCRs

in the pathogenesis of AD and in multiple stages of the processing of APP. Sequential cleavage of APP by the α, β and γ-secretases, which are regulated by GPCRs, determines the extent of amyloid-β peptide generation, and amyloid-β can directly or indirectly affect GPCR function. Specifically, $A\beta_{1-42}$ has been shown to activate specific G-protein coupled receptors, thus affecting the phosphorylation of GEF and GAP receptors, which are related to a death and viability of neuronal cells (Shen et al., 2012). Inhibition of GTPase activity has been shown to protect cells from Aβ induced neurotoxicity (Rymer et al., 2001).

G proteins are known to be involved in signaling via integrins, and are believed to play critical roles in mediating integrin inside-out and outside-in signaling (Shen et al., 2012). The interconnections between these two signaling pathways, the G protein coupled receptor pathway and the integrin pathway, is an example of complex signal integration that would give rise to non-linear dynamics that might be amenable to a systems approach to study. Certainly, to date, most of the analyses have been of single pathways assuming linear relationships, that have only given rise to inconsistencies and apparent competing hypotheses in the literature (Rymer et al.). It is our goal, in this work, to develop tools that will help provide clarity to the hypothesis that Aβ interaction is the basis alters cell homeostasis, at the survival level, when examining complex non-linear pathways in Aβ associated signaling (Asthagiri et al., 2000).

In this work, we will build mechanistic models of Aβ interaction with neurons that focus on the key signaling molecules identified here, integrin, GPCR's, and their ligands, Src and pSrc, and other molecules that could reasonably be assumed to be downstream of such molecules in a signaling network. A detailed network of how integrins, ligands, GPCRs, and Src interact with each other can be seen in Figure 1.1. Additionally, we will utilize stability analyses and sensitivity analyses to describe the perturbations of such biomolecules in the biological system upon addition of Aβ and upon change in model parameters. These tools will help us explore the relative importance of different reactions in the signaling network in model outcome (viability signal), and point to new experiments and new hypotheses, as well as potentially new insights into therapies that could be developed for Aβ related toxicity in disease.

(Figure 1.1: G-Protein model showcasing the interaction between integrins, ligands, GPCRs and Src kinase with respective effector proteins (GEF and GAP), leading to a viability and death signal respectively. (Shen et al., 2012))



Chapter 2: Mathematical Background

Numerical vs Empirical Modeling

Modeling biological systems can be categorized loosely into models that are either primarily data-driven which are empirical in nature, and mechanism-driven models that are depend on a postulated mechanism and experimentally derived rate constants and initial concentration. Data-driven models are constructed solely on analyzing data itself, without having to make any assumptions about the underlying mechanisms. Additionally, they possibly have the advantage of realistically modeling of the biological phenomenon; however, one shortcoming of data-driven modeling techniques is the limited predictive ability in determining the long term effects of cascading biochemical reactions (Steuer, 2007). Data-driven models cannot be used reliably outside of the range of values for which the data was originally collected. When pertinent information such as initial concentrations, rate constants, and mechanistic relationships associated with a signaling cascade is known, as in our case, then the signaling cascade can be modeled specifically with ordinary differential equations (ODEs) that have more predictive capabilities (Steuer, 2007). The mechanistic model also has the advantage of being more easily related to biological reactions that can be directly measured and compared to for validation. Using a mechanistic model, one can then use a variety of computational and numerical tools to examine the effects of perturbations to the system, similar to what a student of chemical engineering would do with an exothermic reaction in a CSTR to determine if a runaway reaction was possible. Past modeling efforts that study the

various signaling cascades in understanding neurotoxicity relevant to AD are scarce. We focus our efforts into developing a numerical model and systems engineering principles to determine the fate of key biological molecules that affect neuron signaling that will impact viability.

Determining Stability of Dynamic Systems

Ordinary differential equations (ODEs), linear or non-linear, can be developed to simulate the fate of system variables as a function of time time as an Initial Value Problem (IVP). Additionally, equilibria points (or steady states in the system) can be determined to examine the stability of the system. There are two points to consider when performing a stability analysis, defining the equilibrium points of the system and the method for determining stability. An equilibrium point of a dynamical system generated by an autonomous system of ODEs is a solution that does not change with time (Hirsch et al., 2004). An equilibrium point can be considered a steady state of the system. In mathematics, stability theory addresses the stability of all possible equilibria points of ODEs and their respective trajectories under small perturbations of initial conditions. In principle, the linear stability of a system is determined by examining the eigenvalues of the system at its equilibria points. For relatively small systems, the equilibria points in the system are solved deterministically and analytically by setting the system of equations to 0; however, with larger systems (# of ODEs > 5), finding equilibria points analytically becomes problematic; equilibria points need to be solved numerically via a root solving algorithm such as the Newton-Raphson method due to the number of variables involved and non-linearity

of the system (Hirsch et al., 2004). Once the equilibria points are found, linear stability at the equilibria points can be determined by examining the eigenvalues, λ, of the set of ODEs.

A negative eigenvalue indicates a linearly stable equilibrium point; that is, after a small perturbation of the system it will return back to the same stable equilibrium point (for chemical reactions, the system will revert to the equilibria concentrations). A positive eigenvalue indicates a linearly unstable equilibrium point; upon a small perturbation of the system, it will move from the unstable equilibrium point (or equilibria concentrations) towards a stable equilibrium point.

Imaginary eigenvalues indicate that the system will oscillate upon perturbation from the equilibrium point.

If the eigenvalues are complex by nature (λ = a+ bi, where i = $\sqrt{-1}$), then the system will exhibit oscillatory behavior either moving towards or away from the equilibrium point, contingent upon the real part of the eigenvalue (Hirsch et al., 2004). Eigenvalues of the system of ODEs are found from the determinant of the Jacobian of the system of ODEs. The table below lists the possible combinations of how both the real and complex parts of eigenvalues affect the stability of the point of interest in a system with two eigenvalues (Hirsch et al., 2004).

Table 2.1: (Describes the nature of the eigenvalue and its relationship with respect to stability in a two variable system (Hirsch et al., 2004).)

Eigenvalues of the Jacobian matrix	Behavior	Stability
real and both positive	source / unstable node	unstable
real and both negative	sink / stable node	asymptotically stable
real and opposite signs	saddle	unstable
complex with positive real part	spiral source	unstable
complex with negative real part	spiral sink	asymptotically stable

To illustrate the method of analysis, we describe a classic stability problem.

Stability Analysis Example: Lotka-Volterra Model

The Lotka–Volterra equations, also known as the predator–prey equations, are a pair of first-order, non-linear, differential equations frequently used to describe the dynamics of biological systems in which two species interact, one as a predator and the other as prey (Shan et al., 2012). The populations change through time according to the pair of equations:

$$\frac{dx}{dt} = \alpha x - \beta xy, \frac{dy}{dt} = \delta xy - \gamma y \qquad (2.1)$$

- x is the number of prey (for example, rabbits);
- y is the number of some predator (for example, foxes);
- t represents time
- $\frac{dx}{dt}$ and $\frac{dy}{dt}$ represent the growth rates of the two populations over time;
- α, β, γ, δ are positive real parameters describing the interaction of the two species.

Additionally, the following assumptions are made about the environment and evolution of the predator and prey populations:

1. The prey population finds ample food at all times.
2. The food supply of the predator population depends entirely on the size of the prey population.
3. The rate of change of population is proportional to its size.
4. During the process, the environment does not change in favor of one species and genetic adaptation is inconsequential.
5. Predators have limitless appetite.

The prey are assumed to have an unlimited food supply, and to reproduce exponentially unless subject to predation; the exponential growth is represented by the term αx. The rate of predation upon the prey is assumed to be proportional to the rate at which the predators and the prey meet; this is represented above by βxy. If either x or y is zero, then there can be no predation. In the predator equation, δxy represents the growth of the predator population while γy represents the loss rate of the predators due to either natural death or emigration; it leads to an exponential decay in the absence of prey (Shan et al., 2012).

Analytically solving for equilibria points

Population equilibrium occurs in the model when neither of the population levels is changing, i.e. when both of the derivatives with respect to time are equal to 0 (Shan et al., 2012).

$$0 = \alpha x - \beta xy, \qquad 0 = \delta xy - \gamma y$$

When solved for *x* and *y*, the above system of equations yields

x	y
0	0
α/β	γ/δ

Determinant of Jacobian: Solving for Eigenvalues

The stability of the fixed point at the origin can be determined by performing a linearization using partial derivatives, while the other fixed point requires a slightly more sophisticated method. The Jacobian matrix of the predator-prey model is

$$J(x,y) = \begin{bmatrix} \partial f/\partial x & \partial g/\partial x \\ \partial f/\partial y & \partial g/\partial y \end{bmatrix} = \begin{bmatrix} \alpha - \beta y & -\beta x \\ \delta y & \delta x - \gamma \end{bmatrix}$$

When evaluated at the steady state of (0, 0) the Jacobian matrix J becomes:

$$J(0,0) = \begin{bmatrix} \alpha & 0 \\ 0 & -\gamma \end{bmatrix},$$

The associated eigenvalues of the equilibria point (0,0) become:

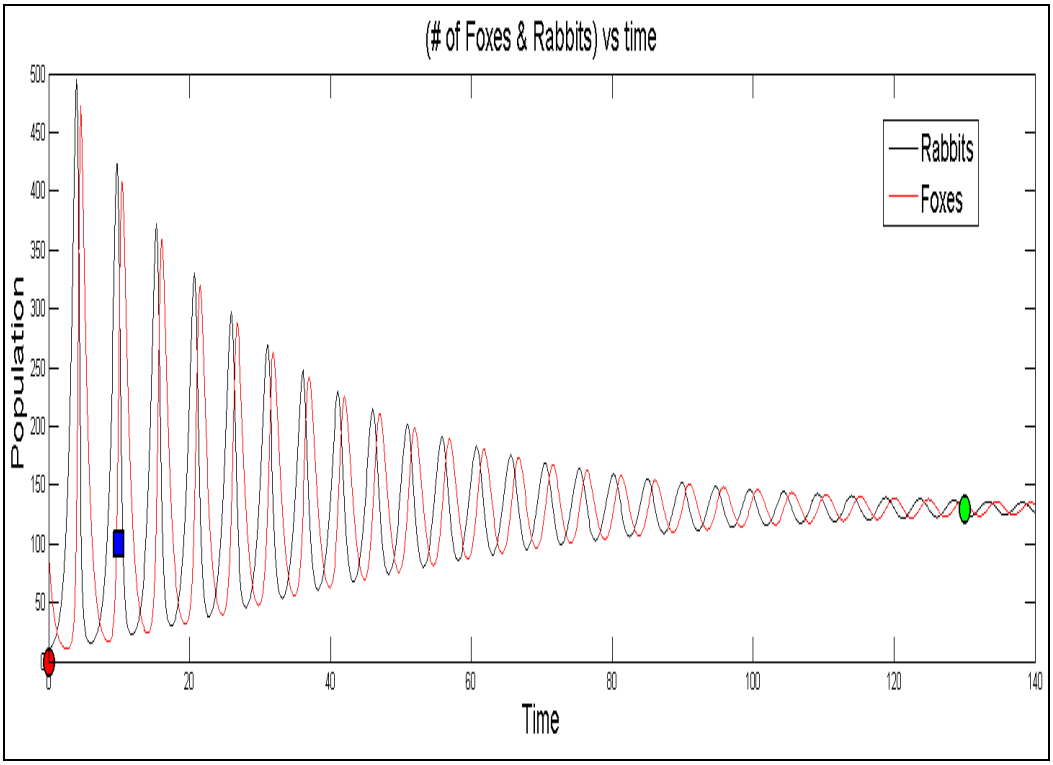
$$det[J(0,0) - \lambda I] \rightarrow det\begin{bmatrix} \alpha - \lambda & 0 \\ 0 & -\gamma - \lambda \end{bmatrix} \rightarrow (\alpha - \lambda)(-\gamma - \lambda) = 0; \quad \boldsymbol{\lambda_1 = \alpha, \lambda_2 = -\gamma}$$

Where, $\lambda I = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$.

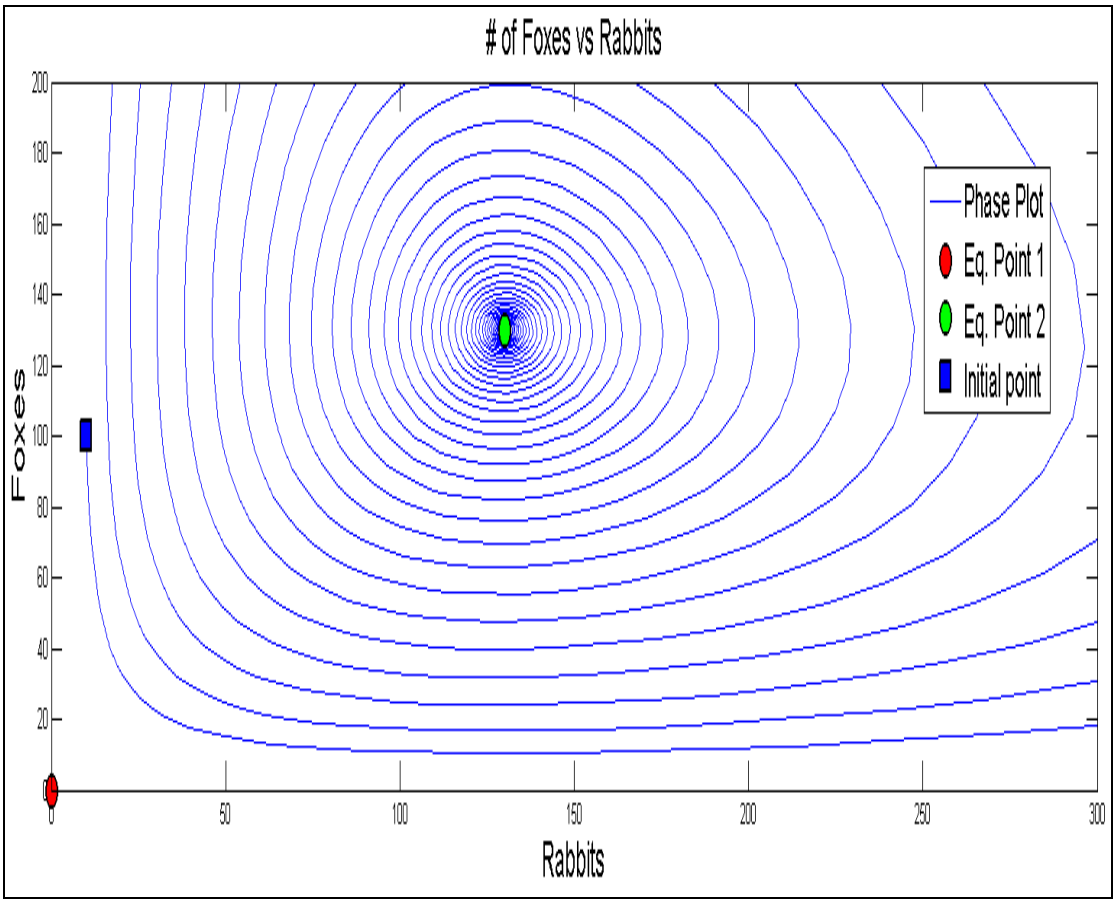
In the model α and γ are always greater than zero, and as such the sign of the eigenvalues above will always differ. Hence the fixed point at the origin is a saddle point. The stability of this fixed point is important since if it were stable, non-zero populations might be attracted towards it, and as such the dynamics of the system might lead towards the extinction of both species for many cases of initial population levels (Shan et al., 2012). However, as the fixed point at the origin is a saddle point, and hence unstable, we find that the extinction of both species is difficult in the model. In fact, this can only occur if the prey are artificially completely eradicated, causing the predators to die of starvation. If the predators are eradicated, the prey population grows unbounded (Shan et al., 2012).

Likewise, solving the second equilibria point $\left(\alpha/\beta, \gamma/\delta\right)$, the eigenvalues are $\boldsymbol{\lambda_1 = i\sqrt{\alpha\gamma}, \lambda_2 = -i\sqrt{\alpha\gamma}}$. As the eigenvalues are both purely imaginary, the fixed point is not hyperbolic, so no conclusions can be drawn from linear analysis. However, as illustrated in Figure 2.1(a), the system eventually reaches a dynamic equilibrium between the number of foxes and rabbits, denoted by 2nd equilibrium point, a green circle. The phase portrait in Figure 2.1(b) exhibits constant orbital motion around the 2nd equilibrium point, which signifies that predator/prey population levels cycle and oscillate around the point. We can use tools like phase plots to illustrate the stability behavior of more complex non-linear systems as well.

(Figure 2.1(a): Fate of rabbits and foxes oscillating with respect to time. Red and green circles represent equilibria points. The blue square represents the initial number of foxes and rabbits.)



(Figure 2.1(b): Phase Portrait of Foxes vs. Rabbits. The phase portrait illustrates the equilibria points. Red and green circles represent equilibria points. The blue square represents the initial number of foxes and rabbits.)



Numerical Based Stability Analysis

Analytical techniques used in solving for stability analysis of dynamic systems are primarily reserved for simplistic linear systems that involve fewer than five ODEs at a time (Kelley, 2003). With that said, non-linearity and higher ordered systems force one to employ numerical methods where initial concentrations and coefficients such as kinetic constants are known beforehand. Similar to employing analytical methods for simpler models, the system of differential equations must be set to 0, $\bigcup_{i=1}^{n} \frac{dx_i}{dt} = 0$, such that the associated equilibria points are numerically found through a root solving algorithm such as the Newton-Raphson method. Once the equilibria points are found, the eigenvalues are found through taking the determinant of the Jacobian,

$$det\left[J\left(x_{1,eq}, x_{2,eq}, \dots x_{n,eq}\right) - \lambda I\right] \rightarrow f(\lambda_1, \lambda_2, \dots, \lambda_n) = 0.$$

Numerical Root Solver for Equilibria: Newton-Raphson Method

In numerical analysis, the Newton-Raphson method is a root finding solver that can be applied to a diverse field of equations, which range from solving a singular equation to a system of equations that can be linear and nonlinear simultaneously (Kelley, 2003). The methodology for a singular equation is as follows: one starts with an initial guess which is reasonably close to the true root, then the function is approximated by its tangent line, which can be computed by taking the differential of the function of interest, and one computes the x-intercept of this tangent line through

elementary algebra. Typically, the x-intercept will be a better approximation to the function's root than the original guess, and the method can be iterated.

In short, the derivation of the method summarizes the steps:

$$y = f'(x_n)(x - x_n) + f(x_n) \to 0 = f'(x_n)(x_{n+1} - x_n) + f(x_n) \to \boldsymbol{x_{n+1}} = \boldsymbol{x_n} - \frac{\boldsymbol{f(x_n)}}{\boldsymbol{f'(x_n)}}$$

Multivariate Newton-Raphson Method

Similarly, the Newton-Raphson method can be applied to a multivariate system with a slight modification to the original method. The system of equations must be differentiated with respect to every variable within the system; as such, an array of independent variables,$[x_1 \ldots x_n]$, and associated functions, $[f_1(x_1 \ldots x_n) \ldots f_n(x_1 \ldots x_n)]$, are utilized to compute the Jacobian of the system to iterate for x_{n+1}. The derivation of the method summarizes the steps:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad f(\boldsymbol{x}) = \begin{bmatrix} f_1(x_1, x_2 \ldots x_n) \\ f_2(x_1, x_2 \ldots x_n) \\ \vdots \\ f_n(x_1, x_2 \ldots x_n) \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \vdots \\ f_n(\boldsymbol{x}) \end{bmatrix}$$

$$\boldsymbol{J} = \frac{\partial(f_1, f_2 \ldots f_n)}{\partial(x_1, x_2 \ldots x_n)} = \begin{bmatrix} \partial f_1/x_1 & \cdots & \partial f_1/x_n \\ \vdots & \ddots & \vdots \\ \partial f_n/x_1 & \cdots & \partial f_n/x_n \end{bmatrix} = \left[\frac{\partial f_i}{\partial x_j}\right]_{nxn}$$, such that

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} - \boldsymbol{J^{-1}} \cdot \boldsymbol{f(x_n)} = \boldsymbol{x_n} - \Delta \boldsymbol{x_n}$$, where $\Delta \boldsymbol{x_n} = \boldsymbol{x_{n+1}} - \boldsymbol{x_n}$.

Even though we have provided examples of how to deterministically solve for eigenvalues analytically and numerically, eigenvalues can also be completely eschewed, favoring entirely different methods such as probability distributions or

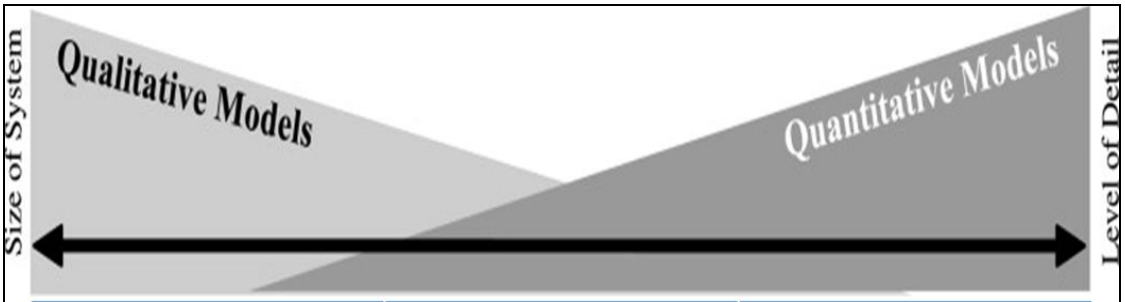
topology in predicting the fate of the variables of the system. Largely, this is dependent on the complexity of the model, mainly on the number of kinetic parameters and variables involved, and how well defined the parameters are. In this work, we focus on using classical linear stability methods of higher order models using numerical techniques. It should be noted, when using analytical techniques to determine stability of a low ordered system, stability criteria can be determined as a function of unspecified parameters, and thus stability as a function of parameter growth rate of prey and death rate of predator can be globally evaluated. However, when using numerical methods for stability determination, only a local solution can be obtained, dependent upon the actual values of parameters (rate constants, initial conditions) used when solving the model. Thus, stability at one set of parameters must be complemented with other methods of analysis, such as sensitivity methods to gather a more complete picture of the dynamic behavior of the system.

A Hierarchy of Numerical Models:

Topology to Detailed Kinetic Modeling

Our goal is to develop a mathematical description of the signaling behavior of Aβ. Based on some estimates, there are over 500 possible signaling reactions in a neuron. Aβ is likely to impact several different receptors, and thus increase the complexity of the signaling network over the "normal" neuron. Generally speaking, including numerous signaling pathways limit the amount quantitative interrogation possible for numerical models as seen in Figure 2.2. Thus, there is a need both to define the scope of the model (how much of the signaling in a neuron to include) and the approach to developing the model.

(Figure 2.2: Spectrum of different types of numerical models. The larger the size of the system, the less quantitative capabilities a modeler has (Steuer, 2009))



Network Analysis	Structured Kinetic Modeling	Kinetic Modeling
Static Description	Dynamic Description	Dynamic Description
No Kinetic Parameters	No Kinetic Parameters	Kinetic Parameters
Topological Properties	Bifurcation Structure	ODEs

Topological Modeling

Due to the inherent limitations of analyzing large-scale kinetic models, topological and graph theoretic approaches have attracted considerable interest (Steuer et al., 2009). In particular, recent advances in genome sequencing and annotation, and thus the possibility to reconstruct large 'genome-scale' metabolic networks for several organisms (Steuer et al., 2009), have triggered an extensive interest in the topological characteristics of metabolic networks. Additionally, topological network analysis has a number of considerable advantages over detailed kinetic modeling, and even Monte-Carlo simulations since it does not presuppose any knowledge of kinetic parameters, thus allowing for an analysis of less well characterized organisms. It is applicable to extensively large systems, consisting of several thousands of nodes, far beyond the realm of current kinetic models; this type of capability allows investigation for a wide variety of topological properties without undue computational effort, such as the degree distribution, average path length, hierarchies and modularity, as well as topological robustness, which contributes to better understanding the metabolic network architecture of said model (Steuer et al., 2009). Nonetheless, an interpretation of metabolic networks solely in topological terms also gives rise to several profound disadvantages when compared to detailed kinetic models. Topological network analysis fails to incorporate the specifically unique properties of a metabolic system such as incorporating initial concentration or rate constant, which are important kinetic parameters in describing the uniqueness of the system; the lack of specifics illustrates a lack of predictive power in determining how biomolecules behave from a time standpoint (Steuer et al., 2009). Despite the

superficial similarities between large classes of biological networks, the structure and function of metabolic systems is fundamentally different from many other networks of cellular interactions.

Since topological modeling fails to address how biomolecules affect one another from a time standpoint, it subsequently fails to provide any meaningful relationships between neurotoxicity and Aβ. As a result, we choose to narrow the scope of our problem by focusing on a select number of key reactions and utilizing more specific modeling techniques such as structured kinetic modeling or detailed kinetic modeling to have a more exact problem to solve.

Structured Kinetic Modeling (SKM)

To allow for an investigation of the structure and function of metabolic systems we have to go beyond merely topological arguments. Structured kinetic modeling (SKM) attempts to combine the advantages of topological based schemes as well as detailed kinetic modeling in the sense of between stoichiometric analysis and explicit kinetic models of metabolism and represents an intermediate step on the way from topological analysis to detailed kinetic models of metabolic pathways The basis of Structural Kinetic Modeling consists of constructing an ensemble of models rather than a single kinetic model, such that the ensemble is consistent with available biological information and additional constraints of interest. From each model, eigenvalues are solved and compiled into a statistical distribution. It provides a fairly detailed analysis of when stability occurs without needing to inherently understand

the kinetics of the system; giving it a distinct advantage over topological modeling (Steuer et al., 2006). However, the drawback is that, depending on the model size, these methods can take more than a concerted effort in determining the accuracy of the eigenvalue distribution. Additionally, the lack of kinetic parameters such as rate constants creates ambiguity of understanding how kinetic parameters affect the stability for all equilibria points involved (Steuer et al., 2006). Hence, we have narrowed the scope of our system to a select number of signaling pathways that we believe are responsible for the propagation of the Aβ, and as a result, utilized deterministic kinetic modeling to design a significantly simpler system with descriptive capabilities (Steuer et al., 2006).

Detailed Kinetic Modeling

As a result, we focused our efforts onto detailed kinetic models since they are the most straightforward and well-known approach to metabolic modeling via ordinary differential equations (ODEs). Similar to other chemical processes, changes in metabolite concentrations are described by a mass-balance equation that incorporates kinetic details of reaction mechanisms and their associated kinetic parameters. Tracing back to the beginning of the last century, detailed kinetic models have contributed significantly to our understanding of the principles of metabolic regulation. In contrast to the situation in many chemical systems, kinetic parameters in biological systems are often context specific. For example, the catalytic activity of enzymes may depend on temperature or pH and other conditions in a complex and nonlinear way. The difficulty to obtain reliable estimates of kinetic parameters is

certainly one of the main hindrances to construct kinetic models on a cellular or compartmental scale (Steuer et al., 2006). Additionally, detailed kinetic models are often constricted to smaller scaled systems or even individual biochemical pathways due to the inherent difficulty in simultaneously solving coupled non-linear ODE for determining eigenvalues and its relationship with their respective equilibria points (Steuer et al., 2006). Nonetheless, the construction of explicit kinetic models allows for a detailed and quantitative interrogation of the alleged properties of a metabolic network – making their construction an indispensable tool of Systems Biology. With a detailed deterministic kinetic model, the stability criterion becomes simpler to determine and evaluate.

We use the same kind of linear stability analysis to address how beta amyloid alters the behavior of signaling networks in neurons and discuss the possible biological interpretations of the results. In short, the eigenvalues determine stability where it is asymptotically stable if all eigenvalues have negative real parts and unstable if at least one eigenvalue has positive real part (Hirsch et al., 2004).

Hence, we describe the development of a deterministic model that describes a subset of the neuronal signaling network that includes relevant parts of pathways that involve Aβ interactions and signal interactions from multiple inputs. We also describe several approaches to examining the stability of the system.

Sensitivity Analysis

A second goal of this work is to examine how the stability of the signaling network in the neuron changes as a function of changes of the inputs to the system. For example, does the stability or trajectory of the behavior of the signaling cascade change with the introduction of an inhibitor to the integrin receptor? Does the stability and/or trajectory change if there are fewer G protein coupled receptors? Does the stability change if the Src kinase is less active? We ask these questions to determine the magnitude of change of kinetic inputs and observe the number of effects it can have on the desired output. These changes could provide insight as to why there may be biological differences in susceptibility to Alzheimer's disease, and what methods of therapeutic intervention might be impactful for AD. Thus, we use sensitivity analysis and vary the initial concentrations of metabolites and rate constants, and observe their effects on the rate and magnitude of the terminal values for the viability and death signal of neurons (Hamby, 1995). Sensitivity analysis aims to describe how much model output values are affected by changes in model input values. In turn, sensitivity analysis can help in identifying critical control points, prioritizing additional data collection, model validation and verification. Furthermore, given the wide magnitude of literature values for kinetic constants and initial conditions, sensitivity analysis can be used to help develop a "comfort level" with a particular model. If the model response is reasonable, after varying model parameters, from an intuitive or theoretical perspective, then the model user may have some comfort with the qualitative behavior of the model even if the quantitative precision or accuracy is unknown (Hamby, 1995).

Different methods are chosen for a variety of reasons, sometimes to attain a quick analysis of the parameters at play without investing an exorbitant amount of time. For example, one of the simplest and most common approaches is that of changing one-factor-at-a-time (OAT), to observe the effect it produces on the output. This appears as a logical approach as any change observed in the output will unambiguously be due to the single variable changed. Furthermore, by changing one variable at a time, one can keep all other variables fixed to their central or baseline values. This increases the comparability of the results since all effects are computed with reference to the same central point in space. Despite its simplicity and ease of use, this approach does not fully explore the input space, since it does not take into account the simultaneous variation of input variables. This means that the OAT approach cannot detect the presence of interactions between input variables (Hamby, 1995).

On the other hand, Statistical Design of Experiments (DoE), a difficult, but more robust and preferred method can be employed to fully examine the input space that affects stability analysis as well as the magnitude of concentrations. When applied to numerical modeling and simulation design, numerous runs are conducted to fully examine the input space and their respective interactions. A conventional and very useful form of DoE is a factorial design, where each factor is assigned a spectrum of values ranging from lowest to highest, dictating the number of runs necessary for sensitivity analysis. This design has been shown to be not only economical but also

effective at revealing interaction effects, and given the non-linearity of biochemical modeling, can give great insight of the interaction effects between parameters, something that OAT and similar methods that fail to do so (Box et al., 2005). Greater detail of the different type of statistical designs is given in Table 2 (Box et al., 2005).

Chapter 3: Methods

The goal of the work presented was to examine the role of different biologically relevant parameters (for example integrin concentration, extracellular matrix concentration, or Src activity) on the behavior of a cell signaling pathway that has relevance to toxicity mechanisms in Alzheimer's using both stability and sensitivity analyses. To accomplish the goal, a mathematical model of the signaling pathway had to be created (described in the results section in Chapters 4 and 5). The model expressed and solved as a series of ordinary differential equations using mass action kinetics, stability determined, phase plots generated to show the relationship between independent variables and model output, and sensitivity plots generated to show the effect on kinetic parameters and initial conditions on model outputs. In this Chapter we describe the algorithms used to generate the results shown in Chapters 4 and 5 for two different signaling model formulations.

MATLAB Solver

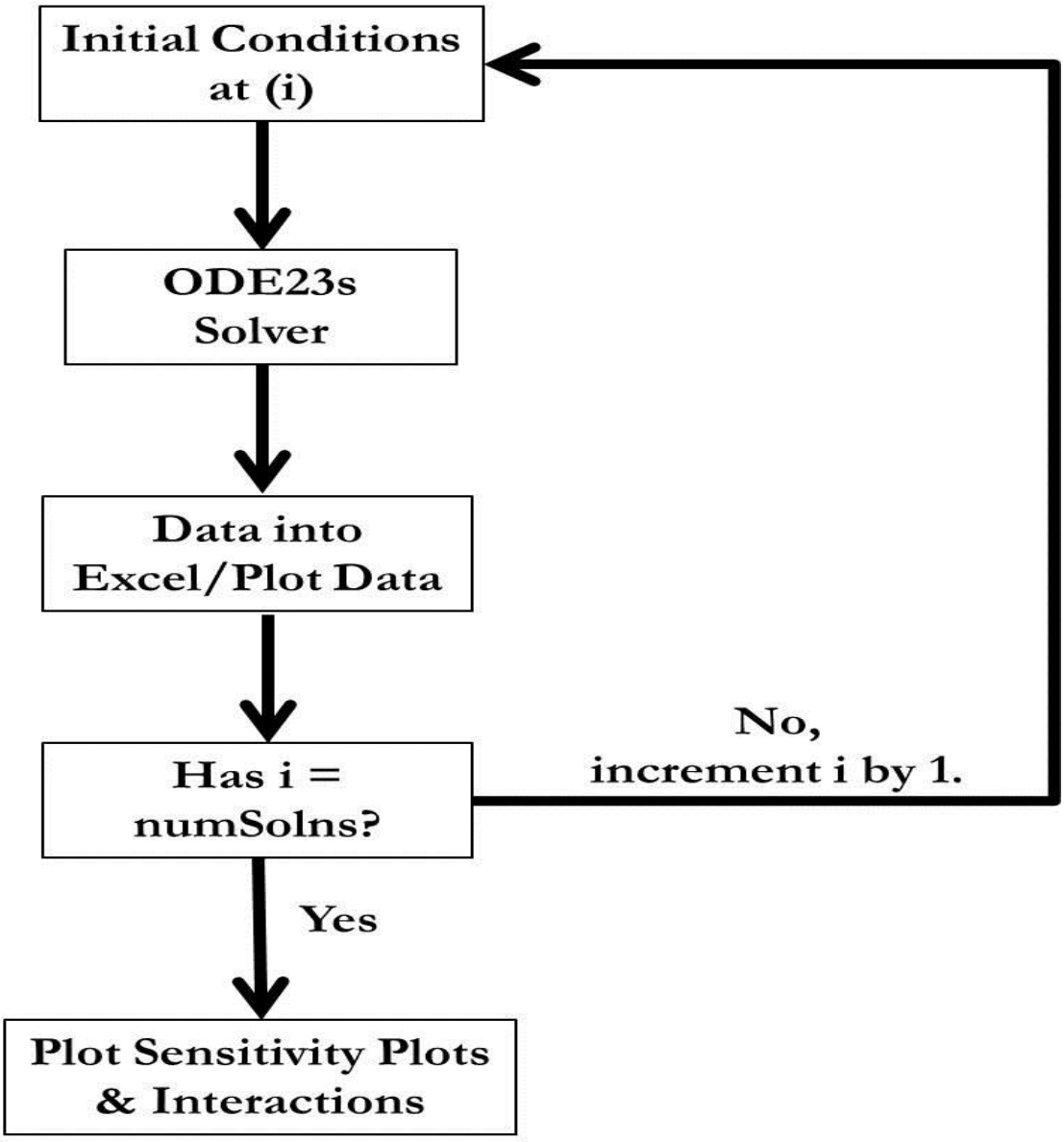
MATLAB was used to solve the set of ordinary differential equations that described each signaling model. Before executing the differential equation solver, a timespan was created with $t_{initial}$ and t_{final} as the starting and ending points. Additionally, initial values were chosen to propagate the solution for the proposed mechanism, where initial concentrations of receptors, ligands, and unactivated intracellular signaling molecule concentrations were set to what was assumed to be reasonable values as

estimated from literature data and/or relative abundance estimates, while activated intermediate concentrations were initially set to zero. Values for initial conditions for each model are presented in tables in Chapters 4 and 5, when the models are presented. Rate constants for all reactions were also estimated from the literature and input as parameters in the differential equation solver. Values for rate constants are presented in Chapters 4 and 5.

To solve the stiff set of coupled ordinary differential equations, a multi-numerical solution procedure was developed with the combination of a counting loop that iterates through "numSolns" number of elements and ode23s, a MATLAB differential equation solver that simultaneously decouples and solves the set of coupled ordinary differential equations. Ode23s is based on a modified 2^{nd} order Rosenbrock formula, which is more efficient than ode45 since it uses cruder tolerances and fewer time steps to solve stiff ODE equations. Within the 1^{st} iteration of the counting loop, assuming that i < numSolns, the initial conditions were passed into a function handle "<insert name here>", which contains the differential and linear equations that mathematically describe the proposed mechanism and its relative rate laws. Once the initial conditions were passed into the function handle, the initial conditions were used to solve for new values at the next time step until ode23s successfully solved for all time steps. After the solution was complete, it was inputted into a solution matrix, which was then graphed and added into a sheet as part of an excel file using the xlswrite command in MATLAB. Finally, the counting loop incremented by one to the next iteration of i+1, and increased the initial conditions to

create a new set of solutions. Once "i" has reached numSolns, interaction plots are created for sensitivity analysis. A flowchart of the process can be seen in Figure 3.1. A copy of the m files used by MATLAB is included in an appendix.

Figure 3.1: (Flowchart of how ODEs are solved via MATLAB, plotted and inputted into an excel file.)



Parameter Estimation

Initial conditions were carefully chosen by comparing literature values from comparable biochemical models that included concentrations of key metabolites: Src, kinases, GPCRs model sets, which ranged from nano- (nM) to micro-molar (μM) (Venkatasubramanian, 2014). Given the complex and uncertain nature in biomolecular modeling, broad ranges for initial concentrations were given, ranging from nanomolar (nM) to micromolar (μM) (Venkatasubramanian, 2014). Subsequently, more specific ranges of metabolites were picked from our lab to validate previous empirical values, and align with the broad ranges found from literature (Venkatasubramanian, 2014). Similarly, kinetic parameters were chosen through a combination of literature values, and validated from our lab. For kinetic constants that could not be validated from literature, half-lives were empirically found (Venkatasubramanin, 2014), and forward kinetic constants were calculated assuming mass-action kinetics. Reverse kinetic constants were scaled relative to forward kinetic constants.

Stability Criterion

Stability criterion was analyzed using a program called COPASI (Hoops et al., 2006). COPASI is open source software used for creating biochemical models and solving those models via various numerical methods. The Simplified and G-Protein model w/Aβ, described in **Chapters 4** and **5**, respectively, were entered into the solver with two sets of initial conditions with the presence and absence of Aβ. Once the values

and mechanism were entered into COPASI, steady-state analysis was chosen in order to calculate the eigenvalues using a combination of solvers including the Newton-Raphson's method (Kelley, 2003). Once the steady state analysis was run, COPASI summarized the results in a report, which contained the number and type of eigenvalue as well as whether an equilibrium steady state was found.

Phase Plots

Once the solver and stability criterion were established for the models, the solver was run in order to generate qualitative trends with respect to time via MATLAB. Phase plots were generated by plotting the values of various biomolecules against the viability signal of the cells. Three different points were chosen from the viability values to denote the starting position (blue circles), unstable equilibrium point (red squares), and the ending values that signify as the stable equilibrium point (green squares) that were identified from COPASI (stability analysis) output. These plots are seen later in Chapters 4 and 5 respectively.

Sensitivity Analysis

We determined the sensitivity of the model to two sets of parameters: initial concentrations and kinetic constants. With a fully defined parameter space, a sensitivity analysis model was designed using a full factorial design via Statistical DoE, factorial design involved observing main and interaction effects of initial concentrations and rate constants. The factorial design was an n^k design, which represents the total number of combinatorial runs needed to fully assess the parameter

space; *k* is the number of parameters analyzed, and each parameter is assessed for *n* number of levels, ranging from the minimum and ending with the maximum (Box et al., 2005). A Yates analysis matrix was generated to assess all possible combinatorial scenarios for each parameter and its associated number of levels (Box et al., 2005); each row represents a set of varied parameters to be entered into ode23s. With respect to initial concentrations, 12 parameters were analyzed at 2 different levels, leading to 2^{12}, or 4096 simulation runs. Likewise, to investigate sensitivity of model output to kinetic constants, 9 parameters that were analyzed at 3 different levels, leading to 3^{10}, or 59049 simulation runs. Once the matrix has been defined, the varied parameters, either the initial conditions or kinetic constants, while the complement set of parameters is kept constant, are entered into ode23s to obtain solution curves and stability points. Afterwards, interaction plots were created to illustrate the magnitude of sensitivity between the stability points for the viability and death signal, and the time needed to achieve stability with respect to the varied parameters. Significance of the changes in model output to changes in parameters was not determined due to challenges with inputting the DoE output into a statistical package (ANOVA). Instead relative importance of a parameter could be inferred from the steepness of the slope of the interaction plot.

Chapter 4: Simplified Model

Model Description

Based on experimental results out of our laboratory (Venkatasubramanian et al., 2014) and results reported in the literature (Berry et al., 1999; Dhawan et al., 2012; Radhakrishnan et al., 2009; Ahn et al., 2010; Ota, 2015), we assumed that there was crosstalk between integrin signaling and G protein coupled signaling within cells that influence cell survival in the presence and absence of Aβ. Therefore, we posited the simplest model we could envision that demonstrated coupling of these signaling networks and influenced unspecified viability (V) or death (D) signals within the cell. We used a generic signaling molecule X* as a mechanism to integrate G protein and integrin related signaling pathways as opposed to positing a specific set of mechanistic reactions that described signal integration between these two pathways. The order of reaction and/or formulation of the mechanism of signal integration is somewhat arbitrary, but satisfied some constraints around a mass balance for the system and our assumptions based on experimental observations (Venkatasubramanian et al., 2014) of the non-linearity of the system.

The Simplified Model includes a ligand- G protein coupled receptor interaction forming an activated complex, described in equation (4.1), where L_1 is the ligand for the G protein coupled receptor (R), and L_1R^* is the activated receptor-ligand complex; and an extracellular matrix ligand (L_2) – integrin (I) interaction to form an activated integrin-ligand complex (L_2I^*)described by equation (4.2). The activated

integrin-ligand complex can then catalyze the activation and phosphorylation Src to form pSrc, described by equation (4.3). Two pSrc can combine to form a downstream activated signal, X* (equation (4.4)). This step, described in equation (4.4) is totally hypothetical, but was consistent with our ideas around the non-linearity of the signaling pathway. The presence of Src leads to the generation of a death signal (D), described in equation (4.5), while the presence of pSrc in combination with viability signals (V) leads to the generation of an additional viability signal, described by equation (4.7). We formulated steps (5) and (7) to be consistent with extracellular matrix (ECM)-integrin signaling leading to a viability signal (that we assume in this case acts through pSrc), and that the absence of the ECM, Src would remain un-phosphorylated and a death signal would be generated, consistent with cell death seen in the absence of ECM-integrin interactions (Hynes et al., 2002). Finally, the activated signal from the pSrc pathway (X*) can interact with the activated G protein coupled receptor- ligand complex (L_1R^*) to lead to additional activation of Src (equation (4.6)).

Reactions included in the Simplified Model

$L_1 + R \overset{K_1}{\leftrightarrow} L_1R^*$ (4.1)

$L_2 + I \overset{K_2}{\leftrightarrow} L_2I^*$ (4.2)

$L_2I^* + Src \overset{K_3}{\leftrightarrow} L_2 + I + pSrc$ (4.3)

$2pSrc \overset{K_4}{\leftrightarrow} X^*$ (4.4)

$V + Src \overset{k_5}{\rightarrow} D$ (4.5)

$L_1R^* + X^* \overset{K_6}{\leftrightarrow} L_1 + R + 2pSrc$ (4.6)

$V + pSrc \overset{k_7}{\rightarrow} 2V$ (4.7)

Rate laws for each reaction were then developed based on the assumption of mass action kinetics. The resultant rate laws are described by equations 4.8-4.14. From the rate laws, a system of ordinary differential equations was developed that were then subjected to further analysis.

Rate Equations: Simplified Model

$r_1 = - k_1.L_1.R + k_{1r}.L_1R^*$ (4.8)

$r_2 = -k_2.L_2.I + k_{2r}.L_2I^*$ (4.9)

$r_3 = -k_3.L_2I^*.Src + k_{3r}.L_2.I.pSrc$ (4.10)

$r_4 = -k_4.pSrc^2 + k_{4r}.X^*$ (4.11)

$r_5 = -k_5.Src$ (4.12)

$r_6 = -k_6.\ L_1R^*.X^* + k_{6r}.L_1.R.pSrc^2$ (4.13)

$r_7 = -k_7.V.pSrc$ (4.14)

In the presence of Aβ, the reactions change as follows. We assume that Aβ can interact with an integrin on the cell surface (I), to form an activated Aβ-integrin complex (Aβ-I*), described by equation (15). This would lead to competitive inhibition of extracellular matrix binding to the integrin receptor, which could be expected to lead to generation of a death signal (or loss of a viability signal), consistent with findings from our laboratory that Aβ leads to toxicity, and inhibition of Aβ integrin binding attenuates toxicity (Venkatasubramanian et al., 2014).

Simplified Model – Aβ

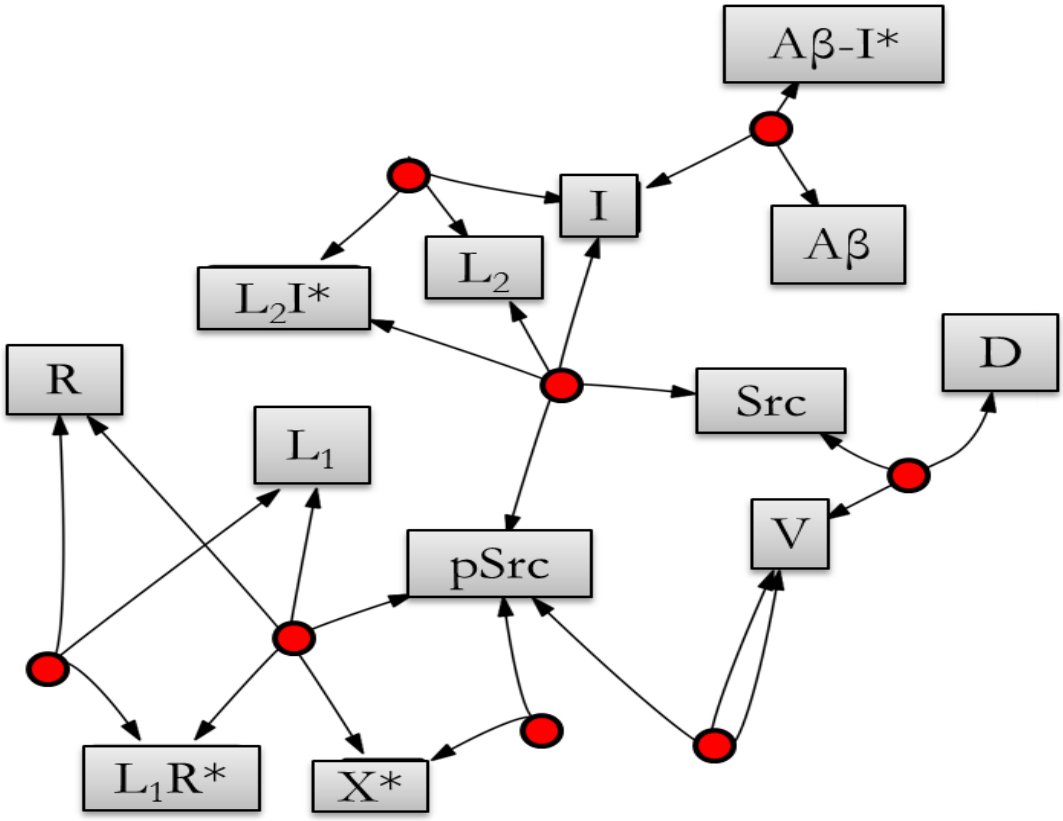
$$A\beta + I \overset{K_8}{\leftrightarrow} A\beta\text{-}I^* \qquad (4.15)$$

The rate law to describe this reaction is as follows:

$$r_8 = -k_8.\ A\beta.I + k_{8r}.A\beta\text{-}I^* \qquad (4.16)$$

A chemical reaction network of the governing equations can be seen in Figure 4.1, which illustrates the crosstalk between ligands, integrins, GPCRs and how Aβ perturbs the system, leading to an increase of a death signal.

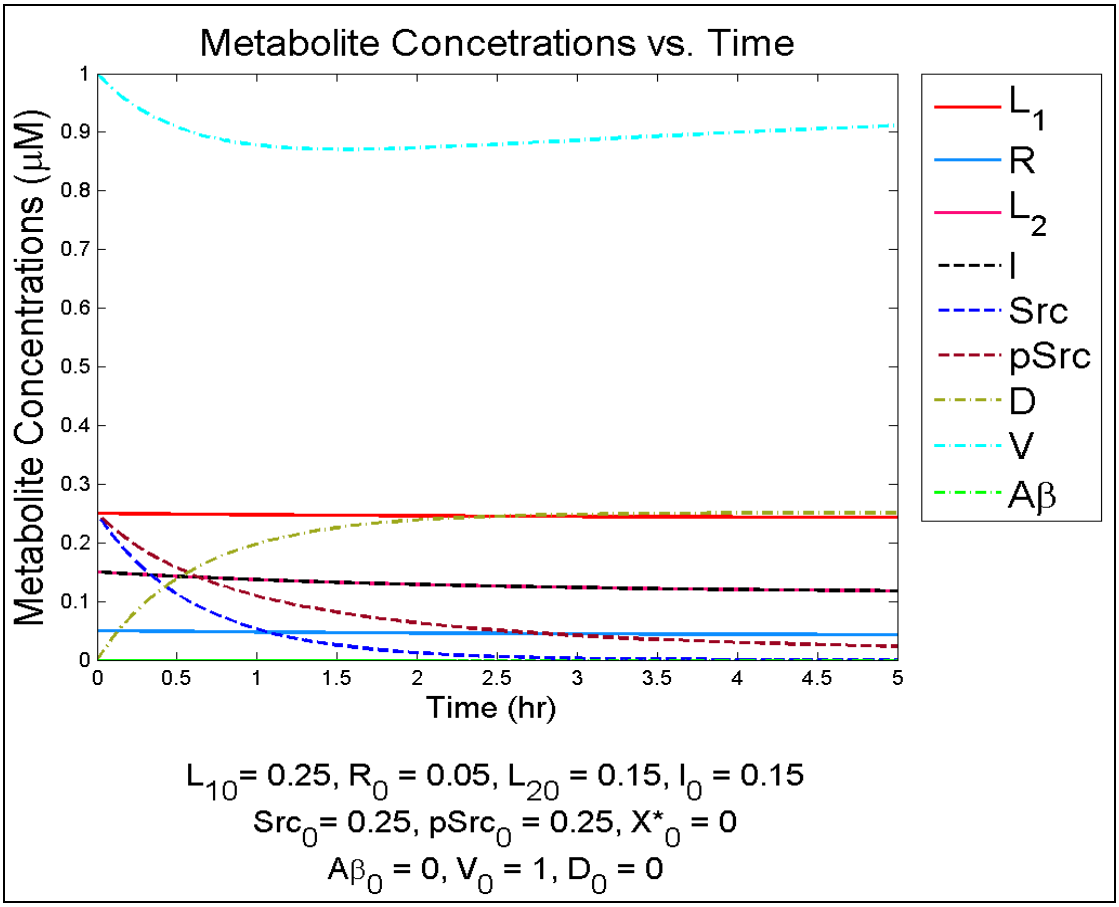
Figure 4.1: (Chemical Reaction Network (CRN) diagram depicting the interactions between key biomolecules, forming activated complexes, and eventually leading to a viability or death signal. Reaction nodes (red circle with black outline) with arrows represent a reversible or irreversible reaction with the biomolecules involved.)



Qualitative Trends

ECM binding to integrin on the cell surface was proposed to generate a survival signal. In this model, the survival of the biological system was believed to be dependent upon the ligand-integrin reaction. G protein coupled receptor (GPCR) - ligand interactions were also included as they contribute to cell survival or death, and likely intersect with integrin triggered signaling pathways. The activated ligand-integrin biomolecules phosphorylate Src, forming pSrc, which is the activated version of the biomolecule. Activated pSrc leads to a dynamic equilibrium between cell death and cell viability signal, which while depicted simplistically in the model, is likely a result of a combination of intracellular signals such as phosphoinositide 3-kinase (PI3K) and Atk and other downstream effectors (Zhang et al., 1996). The graphical trends portrayed by the Simplified model are displayed in Figure 4.2(a).

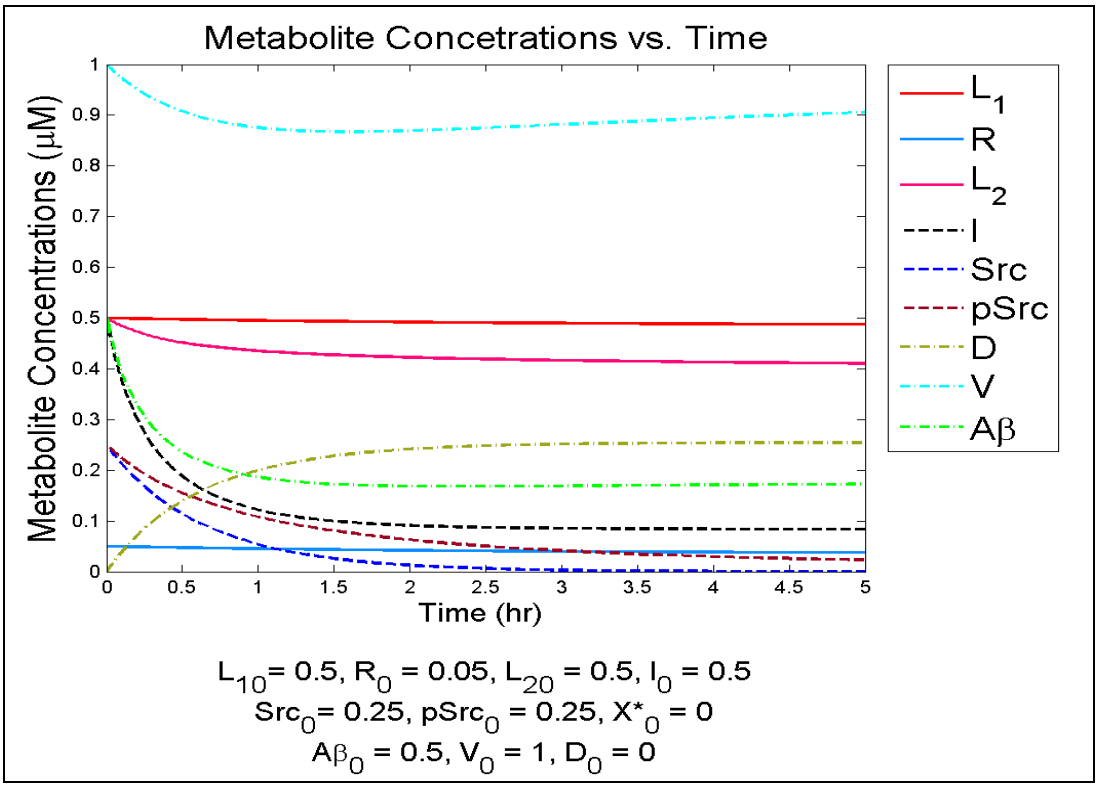
Figure 4.2(a): (Concentrations as a function of time in the Simplified Model with key metabolites (μM) vs. Time (hr) in the absence of Aβ. Global stability is reached quickly and the viability signal is maintained at 1 μM.)



The trends predicted by the model qualitatively describe some of the expected results. The viability signal exceeded the death signal in the control case. The dynamics of the viability signal follow trends observed in the dynamics of the Src and pSrc signals. Neither GPCR Integrin, nor Ligand levels change significantly, which may be a result of the parameters chosen for the model, or the mechanism assumed. Given the lack of available dynamic data on signaling molecule concentration available in the literature, no more than a qualitative examination of the results are possible and their comparison is against expected trends, but not actual experimental data.

In Figure 4.2(b) we show model estimates of metabolites when Aβ is present in the system. ECM binding to integrin on the cell surface was proposed to generate a survival signal. We hypothesized that Aβ plays a significant role in reducing the cell viability signal by competitively inhibited the ECM-integrin interaction. However, inclusion of this step in the model did not yield the expected results. As shown in Figure 4.2(b), the viability signal is still high while the death signal is unchanged relative to Figure 4.2(a).

Figure 4.2(b): (Concentrations as a function of time in the Simplified Model with key metabolites (μM) vs. Time (hr) in the presence of Aβ. Global stability is reached quickly and the viability signal is maintained at 1 μM.)



In addition, while the addition of Aβ led to a decrease in the integrin in the system compared to Figure 4.2(a), the system without Aβ, there was no noticeable change in

Src or pSrc levels, the signaling molecules through which viability and death signals are generated. Another somewhat problematic aspect of this simplified model, we know from experiment, inhibition of pSrc formation attenuates Aβ toxicity (Venkatasubramanian et al., 2014; Rymer et al.), while in this model, pSrc is used to generate a viability signal. Aβ concentrations at 0.175 to 1.4 μM were simulated using the model and the parameter set given in Table 4.1. Changing the concentration of Aβ did not change the qualitative behavior of the model. The mismatch between our expectations of the model could simply be a result of the model parameters chosen for the simulation, or could be a result of the structure of the model and the reactions and their mechanisms included in the model.

To further probe the reason for the failure of the model to at least qualitative capture expected trends in viability signal, we performed both a stability analysis and a sensitivity analysis of the model. The stability analysis would allow us to address if there are qualitative differences in behavior of the model (different stability behaviors) in the presence and absence of Aβ, while the sensitivity analysis allows us to begin to address if parameters were changed, would the model results have been different.

After performance of the stability analysis on the Simplified Model, we note no changes in stability criteria of the model in the presence or absence of Aβ, and that in each case, the model was globally stable and asymptotically approached a result of high viability signal, low death signal. This is further support for failure of this

simplified model. However, as noted earlier, the numerical stability analysis is dependent upon the parameter values used in the model, thus we sought to ensure that even with changes in parameter values, we would still reject this simplified model.

While the model developed predicts the time dependence of concentrations of intercellular signaling molecules, experimentally, we and others generally only report endpoint measurements (i.e. after 24 hours of treatment, pSrc levels increased or cells died), and not dynamic measurements. Thus we were interested in analyzing the predictions of the endpoint or steady state measurements of the model compared to expected trends inferred from our own and literature data. We were also interested in exploring which steps or parameters in the model had the biggest impact on the steady state model predictions as a way of exploring the governing behavior in our system. As a result, a statistical model, involving a full DoE (Design of Experiment) allows us to fully explore the parameter space (initial conditions and kinetic constants) at different values, to determine the parameters that had the biggest impact onto the steady state values and times to reach steady state.

Sensitivity Analysis

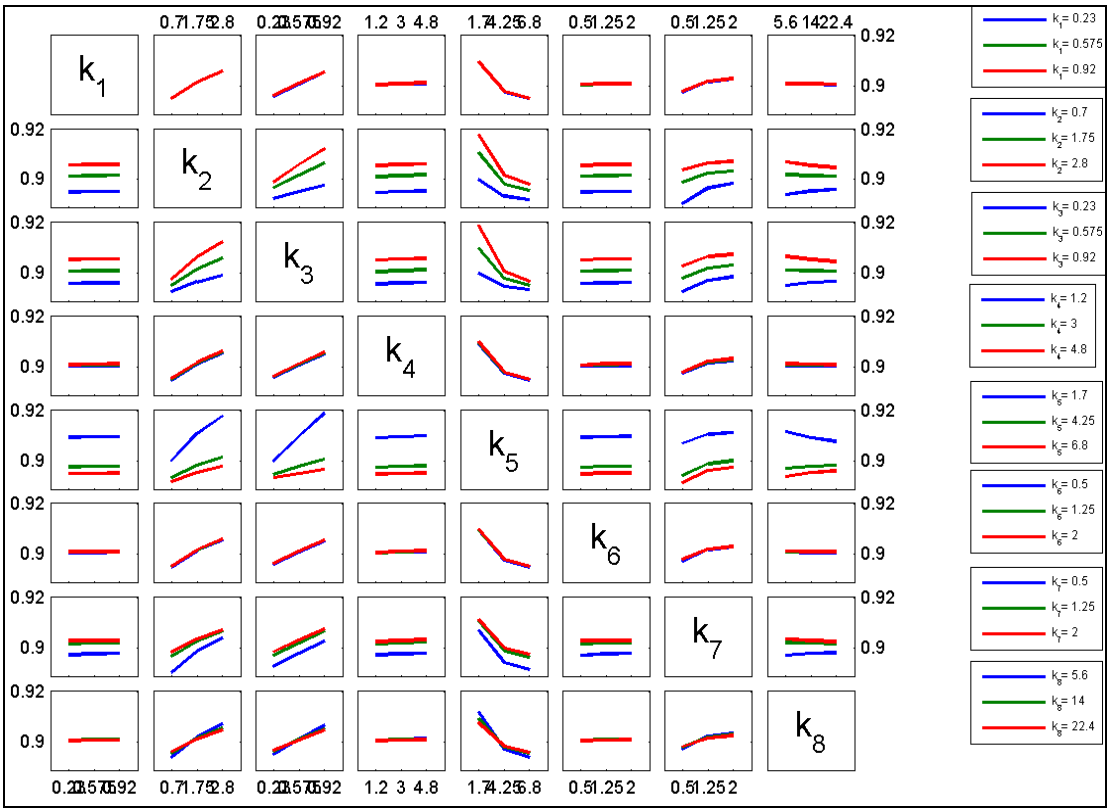
Table 4.1: (Table of Initial conditions and kinetic constants used for stability and sensitivity analysis. Values depicted are base values (i.e. lo w values). High values (HV) are calculated by multiplying the base value (BV) by 4, i.e. $k_{hi} = 4(k_{low})$)

Simplified Model w/ Aβ			
Initial Conditions (μM)		**k**	
L_1 (Berry et al., 1999)	0.5	k_1	0.23 hr^{-1} (half-lives)[1]
R (Berry et al., 1999)	0.05	k_2	0.7 $(\mu M\text{-}hr)^{-1}$ (Welf et al.)
L_2 (Berry et al., 1999)	0.5	k_3	0.23 hr^{-1} (half-lives)[1]
I (Berry et al., 1999)	0.5	k_4	1.2 $(\mu M\text{-}hr)^{-1}$ (scaling)[2]
Src (Dhawan et al., 2012)	0.25	k_5	1.7 $(\mu M\text{-}hr)^{-1}$ (half-lives)[1]
pSrc (Dhawan et al., 2012)	0.25	k_6	0.5 $\mu M^{-2}hr^{-1}$ (scaling)[2]
X* (Radhakrishnan et al., 2009)	0.1	k_7	0.5 $\mu M^{-2}hr^{-1}$ (scaling)[2]
Aβ (Pearson et al., 2006)	0.1	k_8	5.6 $(\mu M\text{-}hr)^{-1}$ (Ahn et al., 2010)
L_1R*	0	k_{1r}	0.25 hr^{-1} (scaling)[2]
L_2I*	0	k_{2r}	0.25 hr^{-1} (scaling)[2]
Aβ-I*	0	k_{3r}	0.25 $\mu M^{-2}hr^{-1}$ (scaling) [2]
D	0	k_{4r}	0.25 $\mu M^{-3}hr^{-1}$ (scaling) [2]
V (scaling)	1	k_{6r}	0.1$(\mu M\text{-}hr)^{-1}$ (scaling)[2]
		k_{8r}	0.1 hr^{-1} (scaling)[2]

In Table 4.1, values of initial conditions and kinetic constants are depicted at the base value for sensitivity analysis. In the following figures, values of either kinetic parameters or initial conditions are varied from their base value (BV) to the high value (HV), such that HV = 4BV, while keeping the other set of parameters constant at their base values, with the exception of Aβ, which is kept at 0.5 μM in order observe if Aβ had noticeable effects onto the viability and death signals as well as steady state times.

[1] Kinetic constant calculated through method of half-lives.
[2] Kinetic constants scaled with respect to forward kinetic constants.

Figure 4.3(a): (Sensitivity plot of steady state viability signals vs. kinetic constants in the presence of Aβ at 0.50 μM. Sensitivity plot depicts main and interaction effects of kinetic constants and their effects onto the viability signal. The steepness of the slope determines the magnitude of effect from kinetic parameters. Since the viability range is from .9 to .92 μM, kinetic constants are shown to have little effect.)



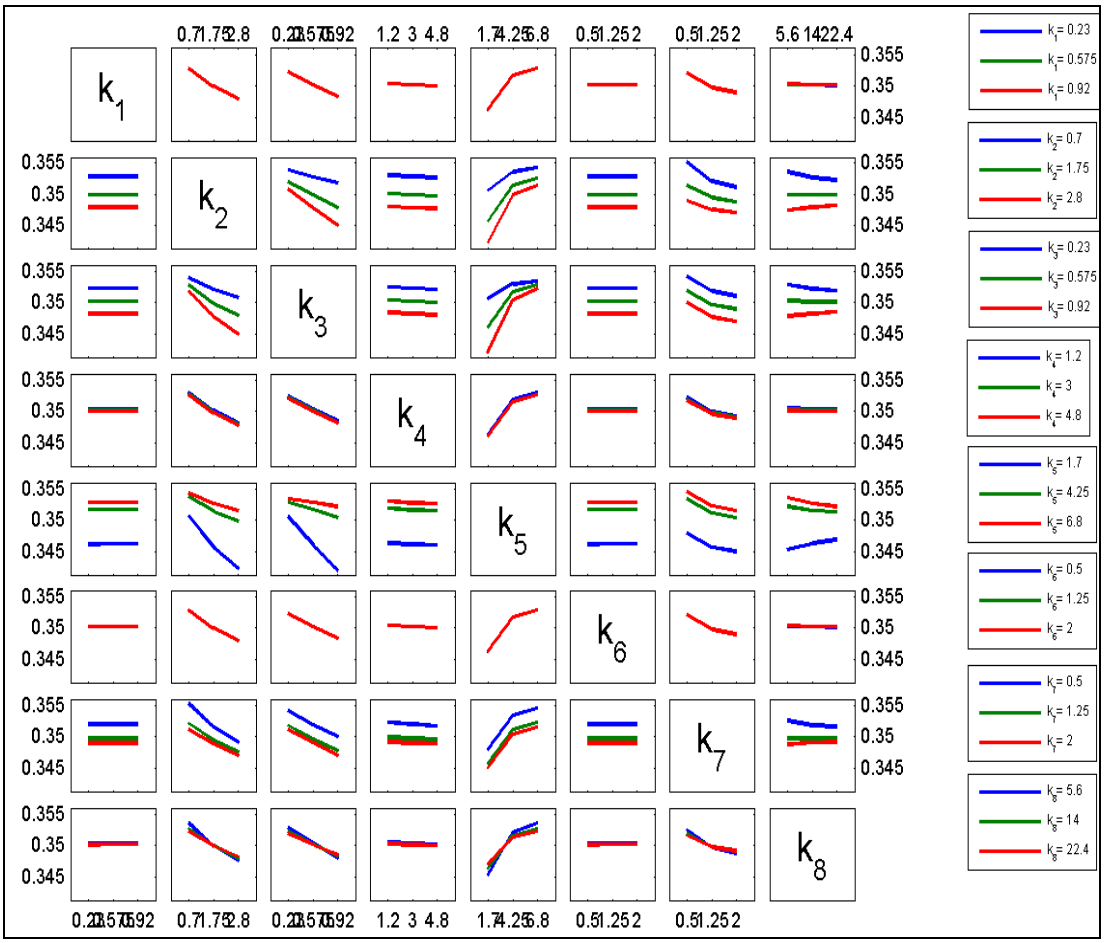
In Figure 4.3(a) the effect of variation in kinetic constants on steady state viability of the model output in the presence of Aβ was shown. In general, steeper slopes indicate a high sensitivity to the rate constant in a given column, while greater distance between the different lines in a plot indicate the sensitivity to the rate constant in a particular row. Kinetic constants k_1, k_4, k_6, and k_8 had little to no effect onto the steady state values for the viability of neurons; while k_2, k_3, k_5 and k_7 had a noticeable effect.

The high sensitivity of model output (steady state viability) to the constants k_2, k_3, k_5 and k_7 is related to the importance of Src and pSrc in this model and their effects on the death (D) and viability (V) signal on neurons. As stated earlier, phosphorylation of Src also leads to regulation of several cellular proteins and receptors, specifically glutamate, NMDA and AMPA receptors, which are related to regulation of LTP and cognitive function. Src and pSrc pathways are also associated with Atk signaling that is generally believed to be associated with survival. Thus, the importance of these steps in survival in the model is consistent with expectations.

On the other hand, k_8, the rate constant associated with Aβ interaction with the cell, had no impact on model output. This suggests either model parameters are very poorly chosen, or that this mechanism is unsuited to describing the effects of Aβ on cell signaling.

From these results we conclude that the Simplified model might be appropriate for describing cell survival reactions associated with integrin and Src/pSrc signaling, but it cannot adequately describe Aβ interactions and their effect on cell signaling, at least for the range of parameters tested in this work.

Figure 4.3(b): (Sensitivity plot of steady state death signals vs. kinetic constants in the presence of Aβ at 0.50 μM . Sensitivity plot depicts main and interaction effects of kinetic constants and their effects onto the viability signal. The steepness of the slope determines the magnitude of effect from kinetic parameters. Since the death range is from .34 to .35 μM, kinetic constants are shown to have little effect.)



In Figure 4.3(b) we show the results of the sensitive analysis where we examined the effect of changing rate constants on the steady state death signals. Again, the same rate constants had the biggest impact on steady state death signals. The two constants that impacted behavior the most, k3 and k5, also show the strongest interaction, where at low values of k_5, changes in k_3 have the largest impact on the steady state death signal value. As in the analysis when the output was the viability signal, the interplay between pSrc and generation of the viability signal and the formation of

pSrc have the strongest impact on model output. Aβ and X* have only minimal input (if any) on model output.

Figure 4.4(a): (Sensitivity plot of steady state viability signals vs. key biomolecules. Sensitivity plot depicts main and interaction effects of molecules and their effects onto the viability signal while kinetic constants are kept at k_{low}. The steepness of the slope determines the magnitude of effect from the parameters. Every biomolecule has shown to be insensitive except for Src and pSrc. However, all parameters effectively have no effect since viability is constant regardless of Aβ's presence.)

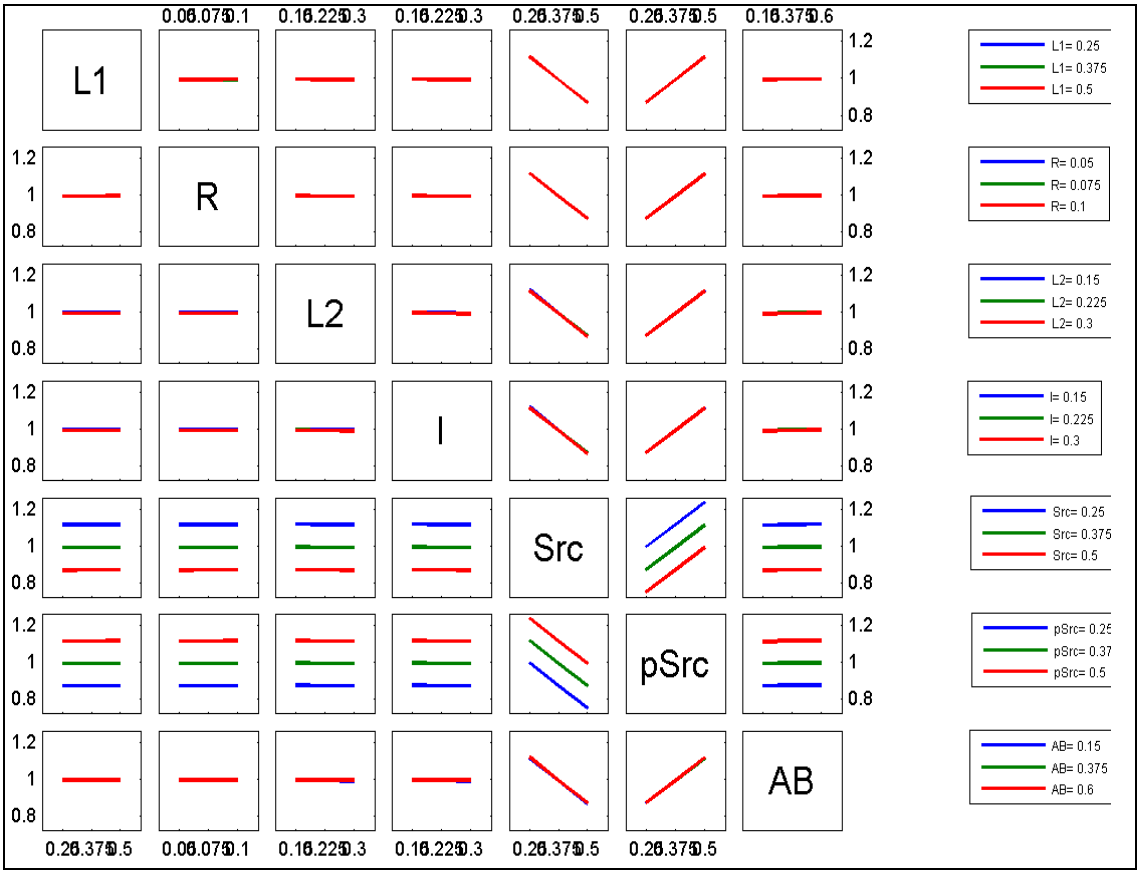
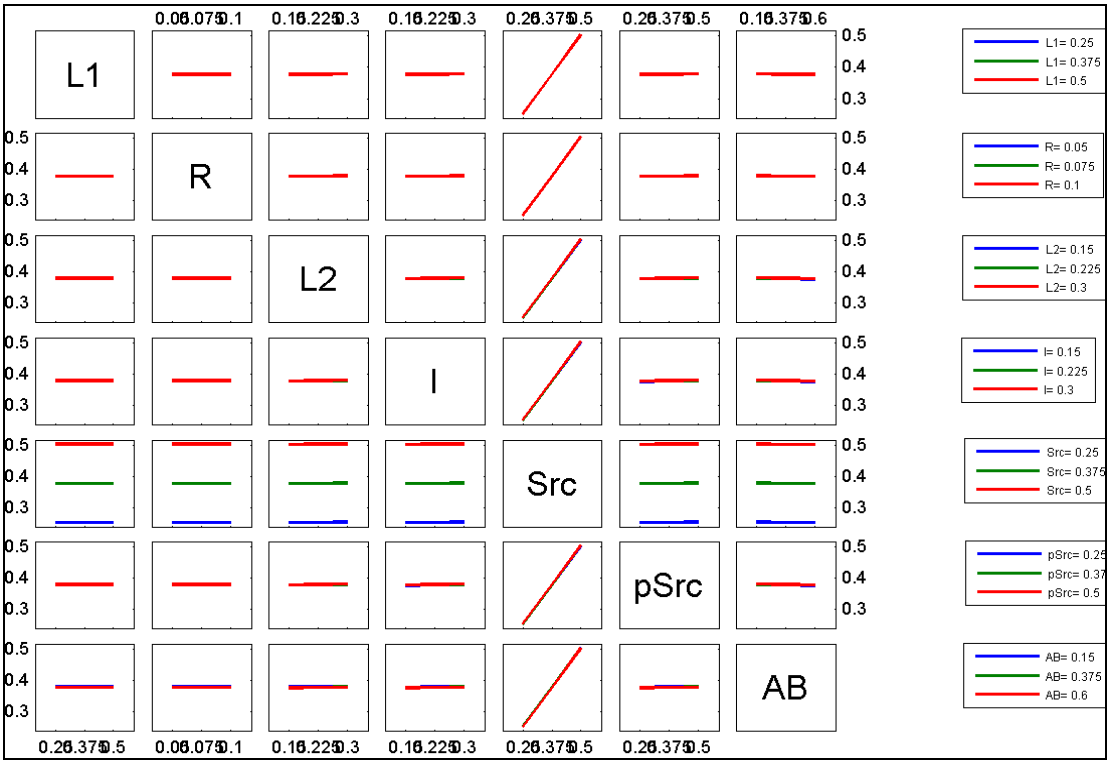


Figure 4.4(b): (Sensitivity plot of steady state death signals vs. key biomolecules. Sensitivity plot depicts main and interaction effects of molecules and their effects onto the death signal while kinetic constants are kept at k_{low}. The steepness of the slope determines the magnitude of effect from the parameters. Every biomolecule has shown to be insensitive except for Src and pSrc.)



Figures 4.4(a) and 4.4(b) show the sensitivity analysis when model predictions were examined as initial conditions were varied, and output of cell viability and death signal, respectively, were observed. As seen in both figures, only initial conditions of Src and pSrc had significant effects on model output, consistent with what was observed when examining rate constants.

Figure 4.5(a): (Sensitivity plot of steady state times vs. kinetic constants. Sensitivity plot depicts main and interaction effects of kinetic constants and their effects onto the steady state times. The steepness of the slope determines the magnitude of effect from the parameters. Kinetic constants k_4 and k_7 have the biggest effect and share interdependence. However, all parameters effectively have no effect since viability is constant regardless of Aβ's presence.)

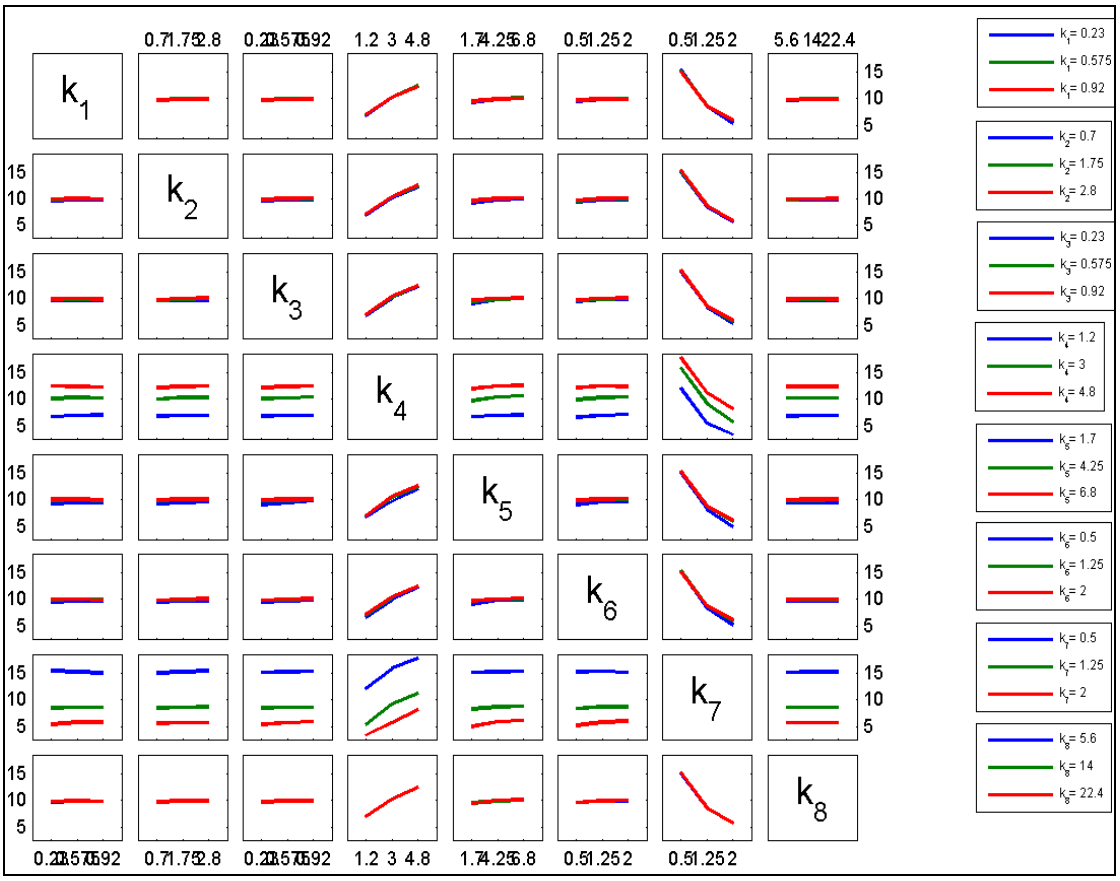
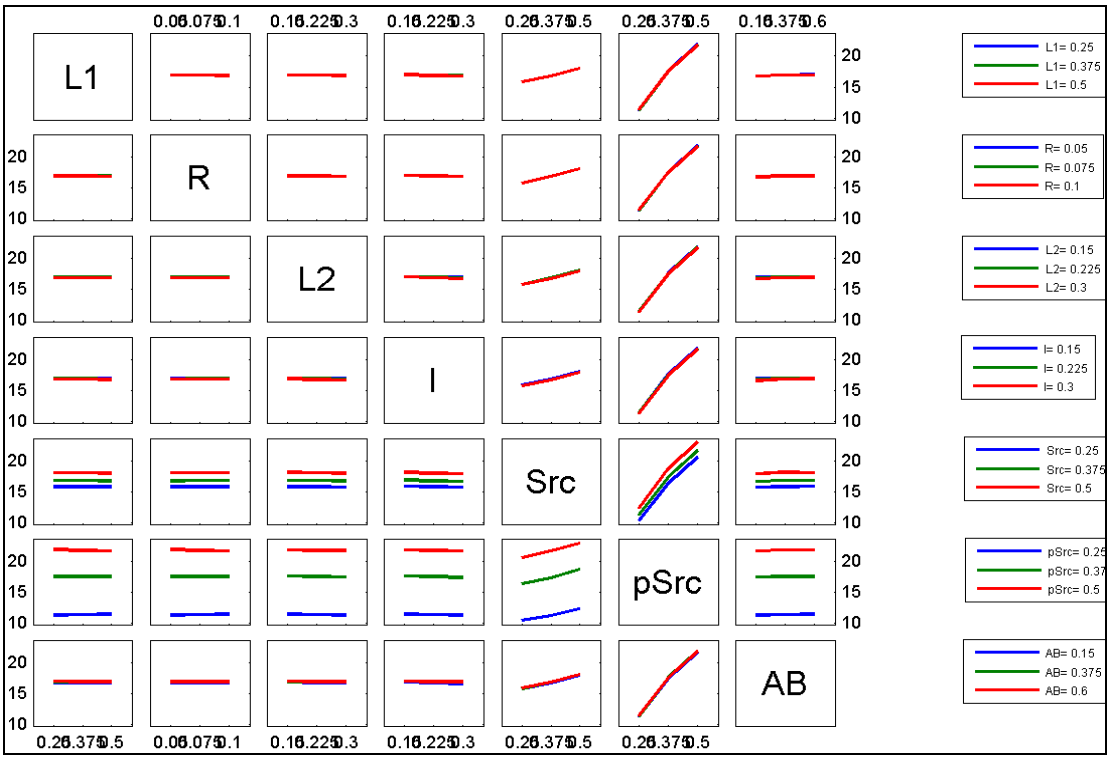


Figure 4.5(b): (Sensitivity plot of steady state times vs. key biomolecules. Sensitivity plot depicts main and interaction effects of the biomolecules and their effects onto the steady state times. The steepness of the slope determines the magnitude of effect from the parameters. Molecules Src and pSrc have the biggest effect and share interdependence. However, all parameters effectively have no effect since viability is constant regardless of Aβ's presence.)



Figures 4.5(a) and 4.5(b) depict how varying initial kinetic constants and initial conditions affect the time needed to reach dynamic equilibrium. In Figure 4.5(a), kinetic constants k_4 and k_7 have the most profound effect on time to reach steady state, while in Figure 4.5(b); pSrc initial conditions are the most impactful. Both rate constants that impact the time to steady state are for reactions that are downstream from the rate constants that had the biggest impact on steady state death and survival signals. The relevance of pSrc on time to steady state, given its importance both in all other sensitivity analysis results and experimental data is not unexpected.

Given the lack of difference in model dynamics in the presence of Aβ compared to in its absence, the lack of difference in any stability measurements of the two models, and the lack of sensitivity of the models to parameters associated with Aβ, there was a need to develop a more complex model of Aβ interaction with the cell signaling network. While the basic signaling associated with integrin, Src, and pSrc might be appropriate in the absence of Aβ, mechanisms that include Aβ must be altered to both capture known literature data and expected trends.

Chapter 5: G-Protein Model

Model Description

The G-Protein model includes many of the biomolecules and reactions from the Simplified Model described in **Chapter 4**, but replaces the downstream signal, X*, with several additional terms for which there is experimental evidence from our laboratory of their role in Aβ signaling, Src kinase and ECM (Venkatasubramanian et al., 2014), and molecules and reactions for which there is evidence from the literature of linkages with pathways in the simplified model. In particular, we added some specific G protein reactions as well as a known mechanism of G protein-Src interactions that might be a possible mechanism of signal integration associated with this system. Equations (5.1) through (5.10) represent this more realistic model of signal integration between integrin, Src and G protein pathways. New biomolecules added include heterotrimeric G-proteins ($G_{\alpha\text{-}GDP}$-$G_{\beta\gamma}$) that disassociate upon interacting with GPCRs, described in equation (5.4). Additionally, in equations (5.6) and (5.7), we included regulatory proteins, GTPase activating protein (GAP) and guanine nucleotide exchange factor (GEF), two effector proteins with the competing functions of stimulating hydrolysis of GTP when bound to a G protein, and the release of GDP and binding of GTP to the G protein, respectively. While GAP and GEF are not known to be involved in Aβ signaling, their inclusions was justified because the role GAP and GEF play in signaling switches in cell phenotype in other systems (spreading versus rounding (Shen et al., 2012)), thus we assumed that they

might play analogous roles in phenotype switching (viability signal versus death signal) in the Aβ signaling system.

Reactions included in the G-Protein Model

$$L_1 + R \overset{K_1}{\leftrightarrow} L_1R^* \quad (5.1)$$

$$L_2 + I \overset{K_2}{\leftrightarrow} L_2I^* \quad (5.2)$$

$$L_2I^* + Src \overset{K_3}{\leftrightarrow} pSrc + L_2 + I \quad (5.3)$$

$$L_1R^* + G_{\alpha\text{-}GDP}\text{-}G_{\beta\gamma} \overset{K_4}{\leftrightarrow} L_1 + R + G_{\alpha}\text{-}_{GTP} + G_{\beta\gamma} \quad (5.4)$$

$$pSrc + G_{\alpha\text{-}GDP} \overset{K_5}{\leftrightarrow} Src + G_{\alpha}\text{-}_{GTP} \quad (5.5)$$

$$G_{\alpha\text{-}GDP} + GEF\text{-}GTP + V \overset{k_6}{\rightarrow} G_{\alpha}\text{-}_{GTP} + GEF\text{-}GDP + D \quad (5.6)$$

$$G_{\alpha\text{-}GTP} + GAP\text{-}GDP + V \overset{k_7}{\rightarrow} G_{\alpha}\text{-}_{GDP} + GAP\text{-}GTP + 2V \quad (5.7)$$

Rate laws for each reaction were then developed based on the assumption of mass action kinetics. The resultant rate laws are described by equations 8-14. From the rate laws, a system of ordinary differential equations was developed that were then subjected to further analysis.

Rate Equations: G-Protein Model

$r_1 = -k_1.L_1.R + k_{1r}.L_1R^*$ (5.8)

$r_2 = -k_2.L_2.I + k_{2r}.L_2I^*$ (5.9)

$r_3 = -k_3.L_2I^*.Src + k_{3r}.\ L_2.I.pSrc$ (5.10)

$r_4 = -k_4.L_1R^*.G_{\alpha\text{-}GDP}\text{-}G_{\beta\gamma} + k_{4r}.L_1.R.G_{\alpha}\text{-}_{GDP}.G_{\beta\gamma}$ (5.11)

$r_5 = -k_5.pSrc.G_{\alpha\text{-}GDP} + k_{5r}.Src.G_{\alpha\text{-}GTP}$ (5.12)

$r_6 = -k_6.G_{\alpha\text{-}GDP}.GEF\text{-}GTP.V$ (5.13)

$r_7 = -k_7.G_{\alpha\text{-}GTP}.GAP\text{-}GDP.V$ (5.14)

In equations (5.15) through (5.17) we include the reactions that describe the interactions of Aβ with signaling pathways in a cell. Upon adding Aβ, we assume that it interacts with an integrin on the cell surface (I), and competitively inhibits the integrin-natural ligand complex (L_2I^*) from forming, which was also a feature of the simplified model. Additionally, Aβ interacts with the heterogeneous trimeric protein to form the GDP bound G protein, causing dissociation of the Gα and Gβγ subunits (reaction 16). Finally, as seen in equation (5.17), activated Aβ-integrin complex (Aβ-I*) phosphorylates Src. The products of both reaction (5.16) and (5.17) lead to an increase of a death signal.

G-Protein Model w/ Aβ (G-Aβ Model)

$A\beta + I \overset{K_8}{\leftrightarrow} A\beta\text{-}I^*$ (5.15)

$A\beta\text{-}I^* + G_{\alpha\text{-}GDP}\text{-}G_{\beta\gamma} \overset{K_9}{\leftrightarrow} A\beta + I + G_{\alpha\text{-}GDP} + G_{\beta\gamma}$ (5.16)

$A\beta\text{-}I^* + Src\ + V \overset{k_{10}}{\rightarrow} pSrc + A\beta + I + D$ (5.17)

Equations (5.8) through (5.14) and equations (5.18) through (5.20) represent the rate expression derived for the reactions shown in equations (5.1) through (5.7) and equations (5.15) through (5.17) assuming the reactions were all elementary and governed by mass action kinetics. At this stage we did not assume any more complex kinetics for any given step in the model. We assume the Aβ-integrin complex will interact with Src, to activate and phosphorylate Src, analogous to the interaction with the integrin-natural ligand complex (L_2I^*); however, when Aβ is in the complex, not only is Src phosphorylated, but a death signal (D) is also generated, as seen in equation (5.17). While there are ample examples of biological reactions that are more accurately described by more complex kinetics (Lee et al., 2007), we have chosen to take the approach that the most simple representation of reactions can still generate interesting non-linear behavior to allow for demonstration of the power of different approaches to examine model properties.

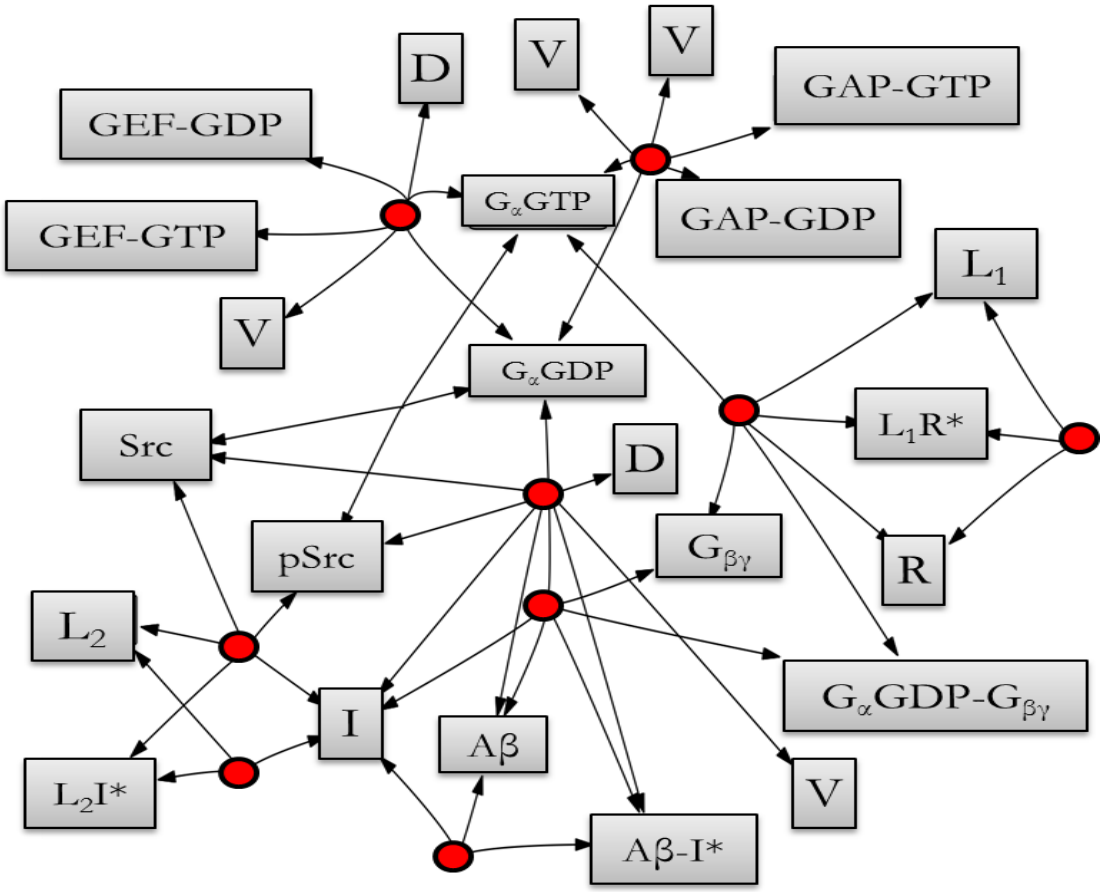
$$r_8 = -k_8.A\beta.I + k_{8r}.A\beta\text{-}I^* \quad (5.18)$$

$$r_9 = -k_9.A\beta\text{-}I^*.G_{\alpha\text{-}GDP}\text{-}G_{\beta\gamma} + k_{9r}.A\beta.I.G_{\alpha}\text{-}_{GDP}.G_{\beta\gamma} \quad (5.19)$$

$$r_{10} = k_{10}.Src.A\beta\text{-}I^*.V \quad (5.20)$$

A chemical reaction network of the governing equations can be seen in Figure 5.1, which illustrates the crosstalk between ligands, integrins, GPCRs and how Aβ perturbs the system, leading to an increase of a death signal.

Figure 5.1: (Chemical Reaction Network (CRN) diagram depicting the interactions between key biomolecules, forming activated complexes, and eventually leading to a viability or death signal. Reaction nodes (red circle with black outline) with arrows represent a reversible or irreversible reaction with the biomolecules involved.)



Qualitative Trends

In Figure 5.2(a) and 5.2(b) we show results of model predictions for the system in which Aβ is absent (the normal, healthy cells signaling case), and in which Aβ is present (the disease signaling case), respectively. While the dynamics of the response of the different signaling molecules is somewhat challenging to interpret (and we later use some other tools that make comparisons of the effects of the model more readily

observable), it is clear that in the presence of Aβ, there are increases in pSrc and the death signal (D) with time. In addition, there appears to be a large decrease in Gα-GDP-Gβγ in the presence of Aβ.

Figure 5.2(a) (Concentrations as a function of time in the G-Protein Model with key metabolites (μM) in the vs. Time (hr) absence of Aβ. Global stability is reached quickly and the viability signal is maintained at 1 μM.)

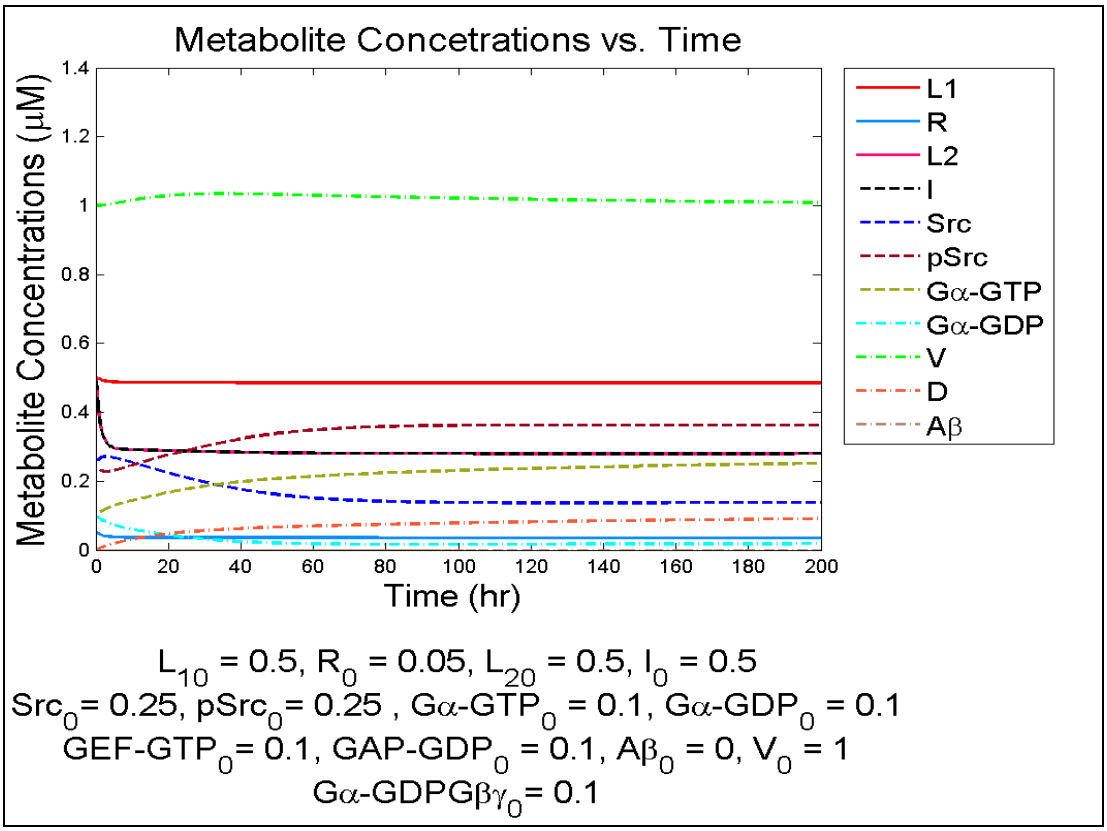
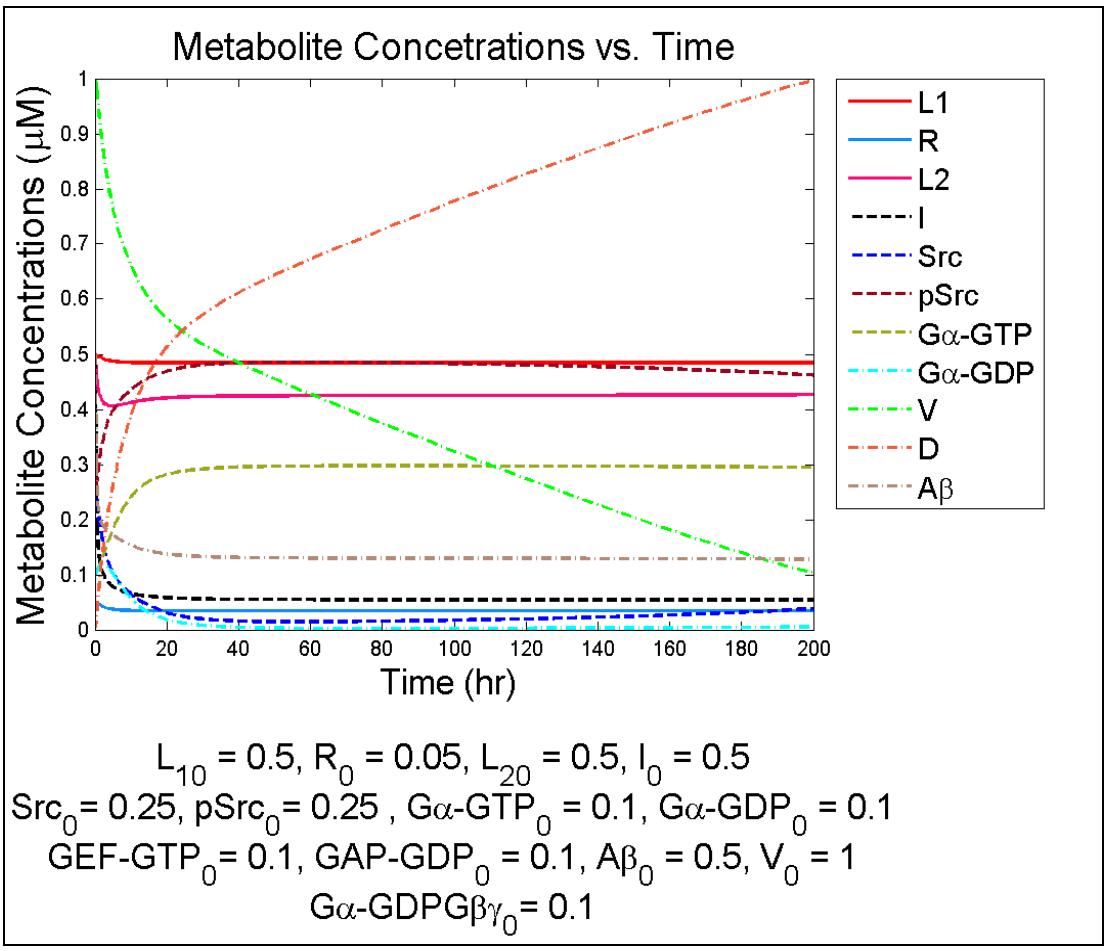


Figure 5.2(b) (Concentrations as a function of time in the G-Protein Model with key metabolites (μM) in the vs. Time (hr) presence of Aβ. Global stability is reached quickly and the viability signal depletes quickly as time progresses.)



ECM binding to integrin on the cell surface was proposed to generate a survival signal, which would be consistent with literature observations (Hynes et al., 2002). In Figures 5.2(a) and 5.2(b), the survival of the biological system was believed to be dependent upon the ligand- G protein coupled receptor reaction, ECM ligand-integrin reaction, and the downstream G protein and Src-pSrc reactions. There are some differences in ligand-G protein coupled receptor and ECM-integrin reactions in the presence and absence of Aβ, seen in Figures 5.2(a) and 5.2(b) , but the impact of

those differences are difficult to see in just a plot of the concentrations as a function of time. The model was also formulated to provide a more realistic picture of the GPCR & G-Protein interaction among neurons while incorporating Src phosphorylation, which is integral for regulation of cell viability (Shen et al., 2012; Ittner et al., 2010). Again, it is difficult to observe the differences in the various G proteins and their effectors, however there are differences in levels of GαGTP between simulations in the presence of absence of Aβ. This could be consistent with observations that Aβ activated a GTPase, and that its inhibition attenuated Aβ toxicity (Rymer et al., 2000). As in other simulations, the lack of Aβ is analogous to a control experiment of examining the survival signal of neurons without any neurodegenerative factors present.

There is virtually no data to compare against in determining the validity of predictions of the model. However, we expect under the "control" condition, without Aβ, shown in Figure 5.2(a) that the cells generally survive, and the interaction with ECM should promote survival. When we examine model predictions in the presence of Aβ, again, there is a lack of experimental data against which to compare the model. Nevertheless, we can examine the qualitative trends in the model compared to some general observations about the impact of certain inhibitors on Aβ toxicity. These observations include that toxicity is attenuated when pSrc formation is blocked, when GTPase activity is blocked, and under some circumstances when Aβ binding to integrin is blocked (Venkatasubramanian et al., 2014; Ittner et al., 2010; Zhang et al., 1996). Consistent with the experimental observations, levels of pSrc and

activated integrin-ligand complex are seen in Figure 5.2(b) compared to 5.2(a), at the same time that more death signal is generated.

We performed a stability analysis of the model to examine the relationship between metabolite or signaling molecule concentrations and viability in the absence and presence of Aβ. Unlike simple systems such as the Lotka-Volterra, in which an analytical stability analysis can be performed, and the nature of the equilibrium states as a function of all rate constants can be determined, in this model, a numerical stability analysis had to be performed. Thus, the solution presented is a function of the parameterization of the model (both rate constants and initial conditions).

Stability Analysis

Phase plots of the viability signal versus various metabolite concentrations are shown in Figure 5.3(a) and 5.3(b) in the absence and presence of Aβ, respectively. The phase plots include a stability well which depicts the global stability between the viability signal within the G-Protein cycle and with other metabolites when Aβ is included. The legend includes the major metabolites that are believed to be responsible for affecting the steady state viability signal. The dependent axis represents the viability signal strength and the independent axis represents the concentrations of species that affect the viability signal. Table 5.1 summarizes the stability criteria for the model.

Figure 5.3(a) (Stability well of Viability versus key biomolecules. Each line represents an individual phase plot between a key biomolecule and the viability signal. Each square represents an equilibrium point, where $\bigcup_{i=1}^{n} \frac{dX_i}{dt} = \mathbf{0}$ ***such that*** X_i ***is a key biomolecule. The red squares represent an unstable equilibrium point while the green squares represent a stable equilibrium point. Blue circles represent the initial condition (shown in Figure 5.3(a), in the absence of Aβ.)***

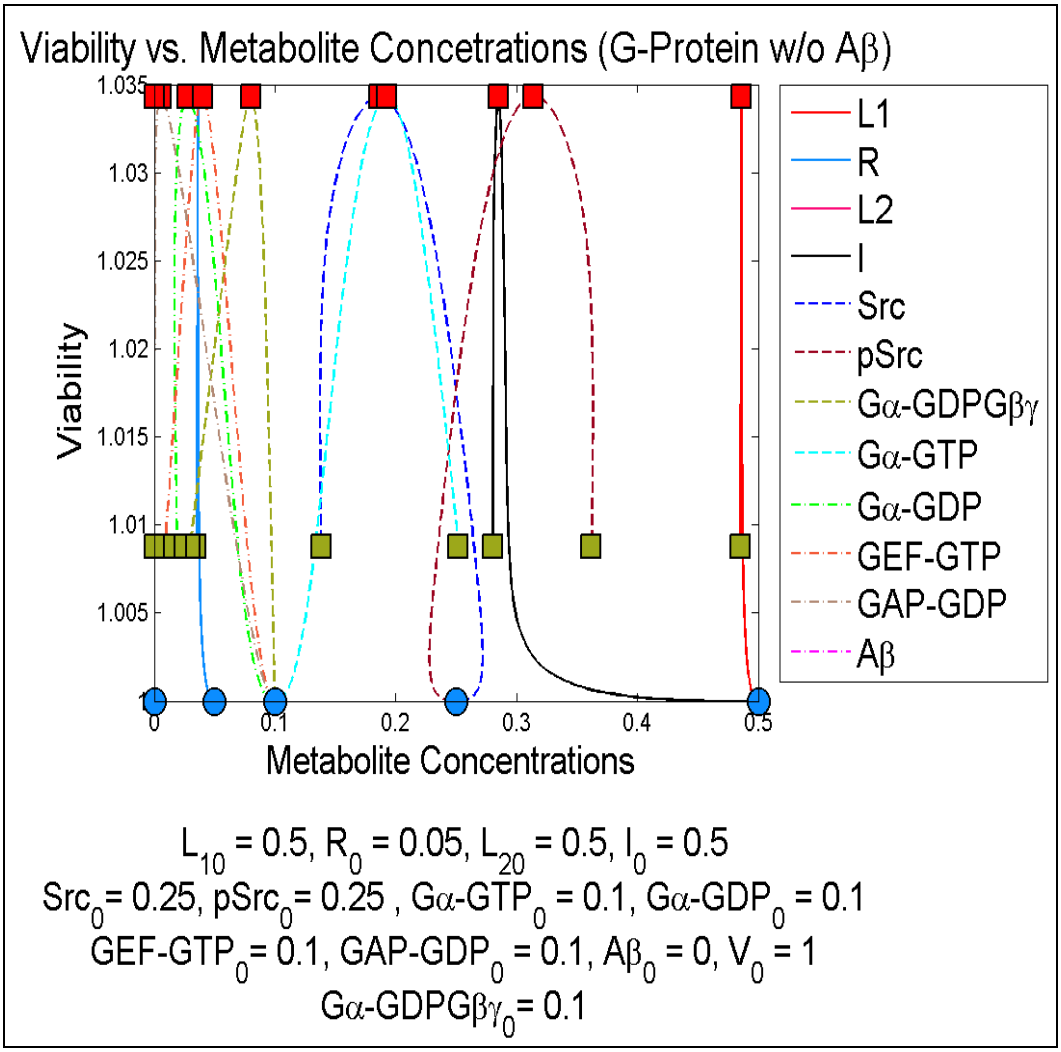


Figure 5.3(b) (Stability well of Viability versus key biomolecules. Each line represents an individual phase plot between a key biomolecule and the viability signal. Each square represents an equilibrium point, where $\bigcup_{i=1}^{n} \frac{dX_i}{dt} = \mathbf{0}$***, such that*** X_i ***is a key biomolecule. The red squares represent an unstable equilibrium point while the green squares represent a stable equilibrium point. Blue circles represent the initial condition (shown in Figure 5.3(b), in the presence of Aβ.)***

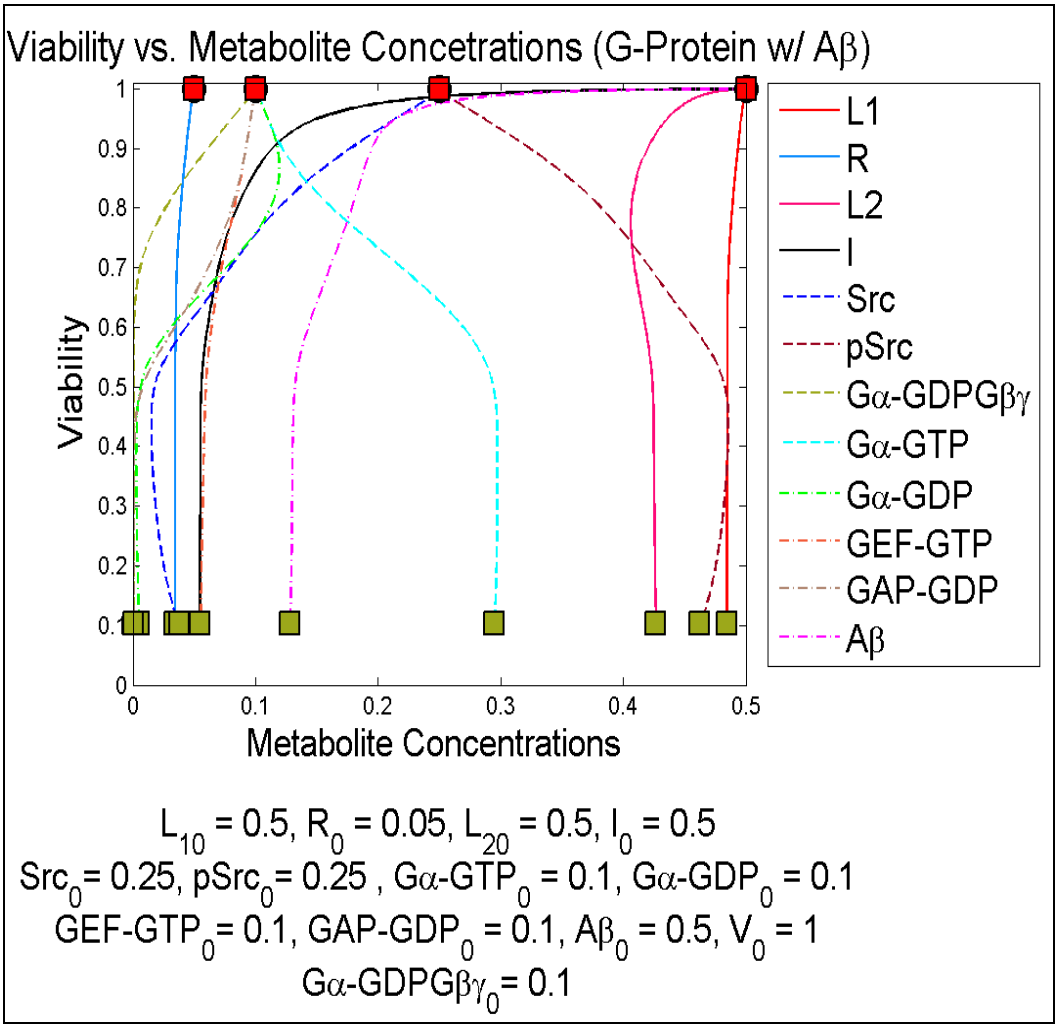


Table 5.1. (Summary of Stability Criteria for Full Model in the absence and presence of Aβ. Real and negative eigenvalues indicate global asymptotic stability.)

Overall Stability of System	**No Aβ**	**w/ Aβ**
	Asymptotically stable	Asymptotically stable
Eigenvalue characteristics		
purely real	10	10
Complex	0	0
equal to zero	0	0
positive real part	0	0
negative real part	10	10

In the absence of Aβ, the steady state of the system as described by the G-protein model (equations 5.1-5.20) represents a global stability; the system behavior tends asymptotically toward one stable state, in this case leading to full viability signal (Figure 5.3(a)). However, with the inclusion of Aβ, the steady state of the system also represents a global stability; but the system behavior tends toward zero viability. In both cases, as seen in the much simpler stability analysis shown in Chapter 2, a system with all real, negative eigenvalues is a locally stable at each equilibrium point. As seen in Figure 5.3(b), all of the stable equilibrium points, while at different concentrations of signaling species, all occur at zero viability signal.

As seen in the much simpler stability analysis shown in Chapter 3, a system with all real, negative eigenvalues is a locally stable at each equilibrium point. As seen in Figure 5.3(b), all of the stable equilibrium points, while at different concentrations of signaling species, all occur at zero viability signal. We speculate that this tendency to asymptotically approach zero viability signal (or death) in the presence and of Aβ is due to the inherent instability of the system during disease (everything always dies). The non-zero viability signal in the absence of Aβ represents a different state of the system – a "healthy state". It is important to emphasize that this result is highly

dependent upon the parameterization of the models, and that there may be a set of parameters in which different local or global stability are observed in the two cases.

Given the dependence of both the dynamic trends predicted by the model and the stability analysis on model parameterization, we performed a sensitivity analysis to determine how modifying rate constants, equilibrium constants, and initial conditions affected model outcome. We included Aβ in the model in the analysis. We focused on 4 model outcomes, the value of the viability and death signals at some time not close to equilibrium (as at equilibrium since everything tends towards zero viability signal the results are uninformative), and the time to achieve the steady state values of the viability and death signals. For each parameter, we varied its value by a factor of 4, and interrogated the effect on model output. We varied parameters two at a time such that interaction effects could be seen. If only a single line is observed in the matrix, then the model is insensitive to that parameter. If horizontal lines are seen in the matrix, then the two parameters probed do not display interaction effects at the ranges of parameters chosen. If, however, multiple diagonal lines are observed, then the model is sensitive to the parameter, and there is an interaction between the two parameters interrogated. Base values for the statistical DoE used for sensitivity analysis can be seen in Table 5.2.

Table 5.2: (Table of Initial conditions and kinetic constants used for stability and sensitivity analysis. Values depicted are base values (i.e. lo w values). High values (HV) are calculated by multiplying the base value (BV) by 4, i.e. $k_{hi} = 4(k_{low})$)

G-Protein Model w/ Aβ			
Initial Conditions (μM)		**K**	
L_1 (Berry et al., 1999)	0.5	k_1	0.23 hr^{-1} (half lives)[1]
R (Berry et al., 1999)	0.05	k_2	0.7 $(\mu M\text{-}hr)^{-1}$ (Welf et al.)
L_2 (Berry et al., 1999)	0.5	k_3	0.23 hr^{-1} (half lives)[1]
I (Berry et al., 1999)	0.5	k_4	0.5 $(\mu M\text{-}hr)^{-1}$ (half lives)[1]
Src (Dhawan et al., 2012)	0.25	k_5	0.5 $(\mu M\text{-}hr)^{-1}$ (scaling)[2]
pSrc (Dhawan et al., 2012)	0.25	k_6	0.5 $\mu M^{-2}hr^{-1}$ (scaling)[2]
Gα-GDP-Gβγ (Radhakrishnan et al., 2009)	0.1	k_7	0.5 $\mu M^{-2}hr^{-1}$ (scaling)[2]
Gα-GTP (Ota, 2015)	0.1	k_8	5.6 $(\mu M\text{-}hr)^{-1}$ (Ahn et al., 2010)
Gα-GDP (Ota, 2015)	0.1	k_9	1.2 $(\mu M\text{-}hr)^{-1}$ (scaling)[2]
GEF-GTP (Radhakrishnan et al., 2009)	0.1	k_{10}	1.2 $\mu M^{-2}hr^{-1}$ (scaling)[2]
GAP-GDP (Radhakrishnan et al., 2009)	0.1	k_{1r}	0.25 hr^{-1} (scaling)[2]
Aβ (Pearson, 2006)	0.2	k_{2r}	0.25 hr^{-1} (scaling)[2]
L_1R^*	0	k_{3r}	0.25 $\mu M^{-2}hr^{-1}$ (scaling)[2]
L_2I^*	0	k_{4r}	0.25 $\mu M^{-3}hr^{-1}$ (scaling)[2]
Gβγ	0	k_{5r}	0.1$(\mu M\text{-}hr)^{-1}$ (scaling)[2]
Aβ-I*	0		
D	0	k_{8r}	0.1 hr^{-1} (scaling)[2]
GEF-GDP	0		
GAP-GTP	0	k_{9r}	0.1 $\mu M^{-3}hr^{-1}$ (scaling)[2]

[1] Kinetic constant calculated through method of half-lives.
[2] Kinetic constants scaled with respect to forward kinetic constants.

Sensitivity Analysis

Figure 5.4(a): (Sensitivity plot of viability signal values vs. kinetic constants at t = 10 hrs. Sensitivity plot depicts main and interaction effects of kinetic constants and its effects onto the viability signal. The steepness of the slope determines the magnitude of effect from the parameters. Kinetic constants k_2, k_3 and k_{10} have the biggest effect and share interdependence. Due to Aβ's presence, viability signal rapidly depletes such that steady state values are effectively 0 µM.)

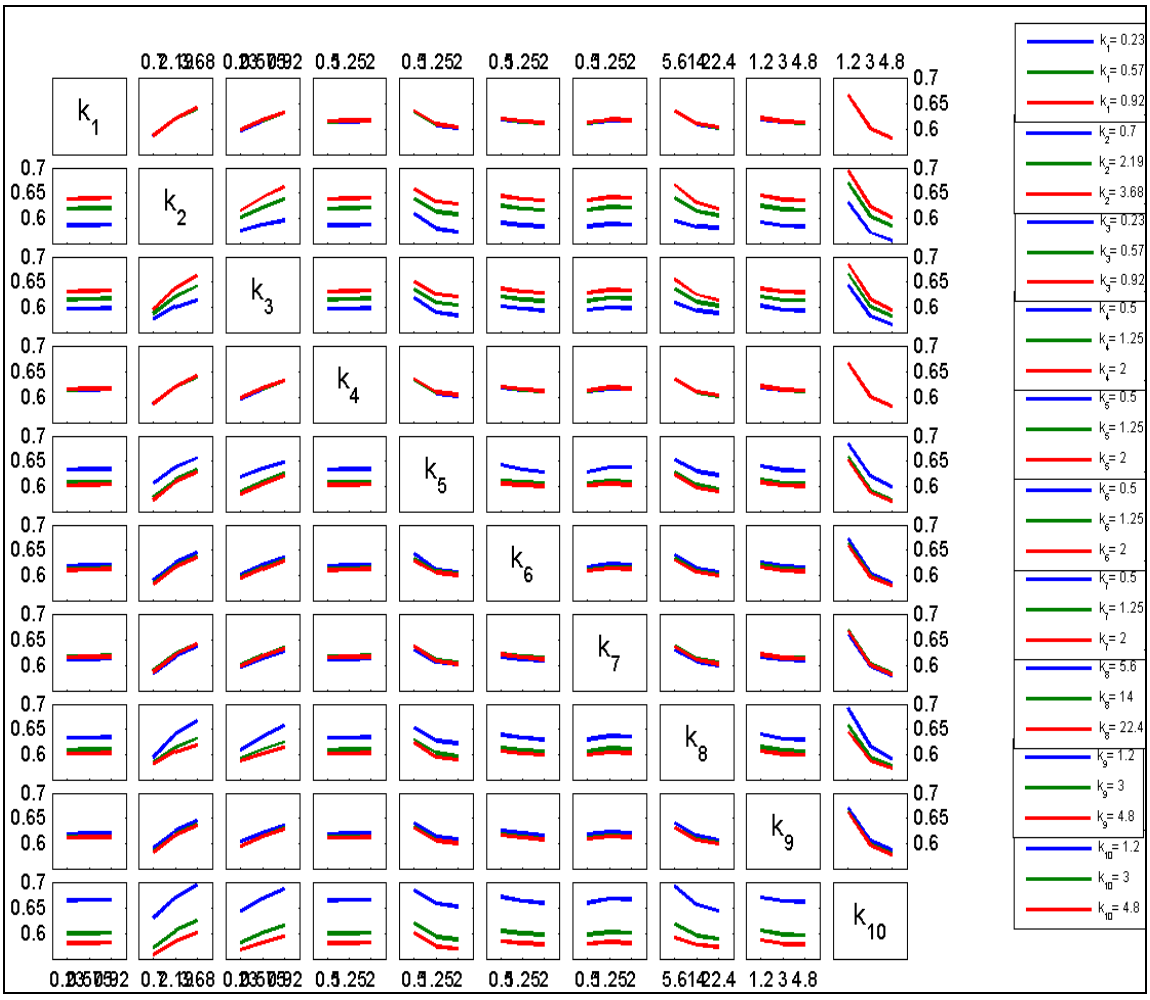
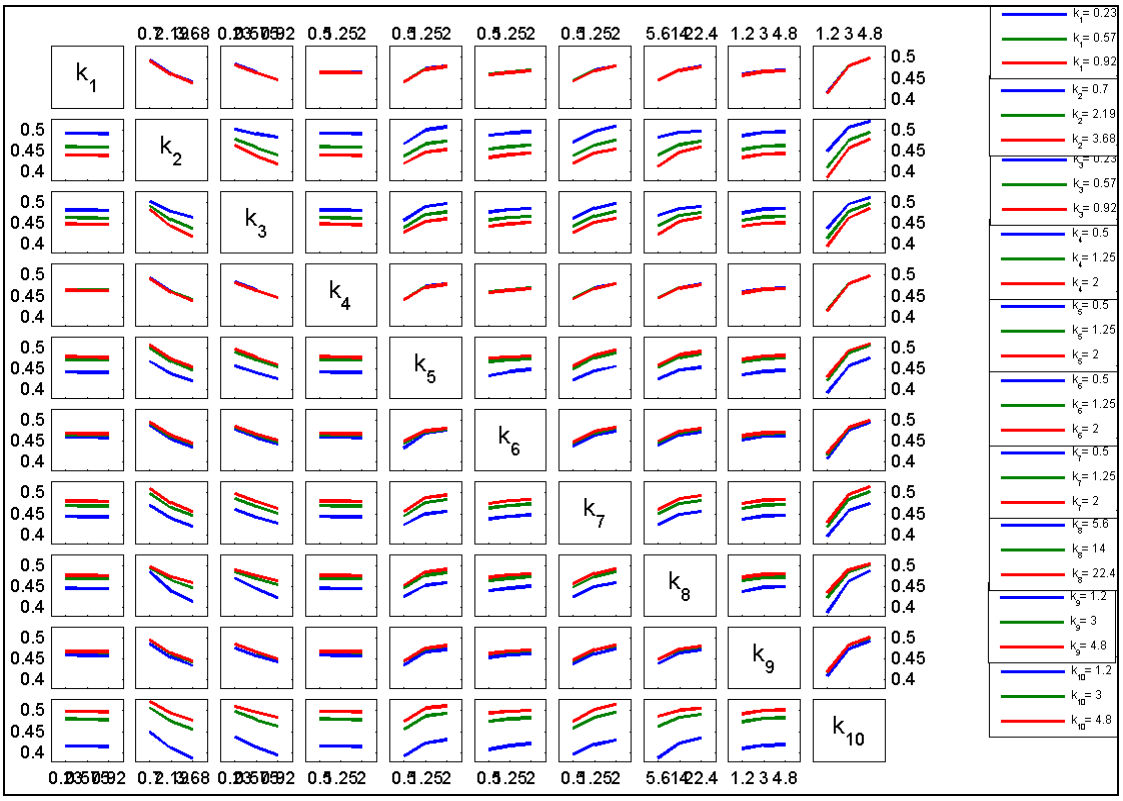


Figure 5.4(b): (Sensitivity plot of death signal values vs. kinetic constants at t = 10 hrs. Sensitivity plot depicts main and interaction effects of kinetic constants and its effects onto the viability signal. The steepness of the slope determines the magnitude of effect from the parameters. Kinetic constants k_2, k_3, and k_{10} have the biggest effect and share interdependence. Due to Aβ's presence, death signal rapidly accumulates such that steady state values are effectively 1 μM.)



Figures 5.4(a) and 5.4(b) depict the sensitivity analysis; the impact of changes in kinetic constants on the viability and death signals, respectively. Forward rate constants k_2, k_3, k_5, k_8, and k_{10} had the most noticeable effects on both the viability and death signal. In addition, k_7 had a noticeable effect solely on the death signal.

The high sensitivity of the constants k_2, k_3, k_8, and k_{10} on model output (viability and death signals) points to the relative importance of the integrin and src pathways in the model. The results also point to the relative insensitivity of the model output to the G

protein coupled receptor pathway. This differs from the sensitivity analysis of the simplified model described in **Chapter 2**, where forward rate constant, k_2, involving L_2 (ECM), had little effect in mitigating the effects of Aβ. The sensitivity of the model to k_2 is consistent with data from our laboratory showing that blocking the Aβ – integrin interaction attenuated Aβ toxicity (Venkatasubramanian et al., 2014), and work by others showing the viability inducing effects of interaction with ECM and 3D culture (Gilmore et al., 2000). Similarly, viability and death signal increase and decrease respectively as k_3 increases. Once the ECM-integrin complex is formed, it activates Src via phosphorylation, forming pSrc. In the absence of Aβ, pSrc acts via a GAP reaction to produce a viability signal. Additionally, phosphorylation of Src helps disassociate the ECM-integrin complex, forming free ECM and integrin to further competitively inhibit Aβ from forming an activated complex. However, both viability and death signal decrease and increase respectively when kinetic constants, k_8 and k_{10} increase. Increased magnitudes of k_8 lead to increased conversion of Aβ into its activated form, Aβ-I*. The activated form has been known to compete with the ECM-Integrin reaction, preventing the GTP-bound form G-protein and GAP reaction, effectively decreasing the viability signal. Additionally, increased values of k10 leads to increased phosphorylation of Src due to activated Aβ-I*, which we propose is responsible for the cell death signal. Experimentally, we have shown that inhibition of Aβ induced pSrc formation attenuates Aβ induced toxicity (Venkatasubramanian et al., 2014), consistent with the role of k_{10} in determining cell fate (viability or death). Finally, the viability increased and death signal decreased as k_5 increased, though to a lesser extent than the aforementioned kinetic constants. The

reaction involving k_5 plays a dual role into forming the GTP-bound G protein, which is we propose is responsible for cell viability; however, the reaction also de-phosphorylates pSrc, creating Src, which is able to be phosphorylated again by the more promiscuous Aβ.

As previous literature has shown, integrin adhesion to extracellular matrix is required for survival of most normal cells (Gilmore et al., 2000) and the survival signal is generated via kinase activity, the first step being phosphorylation of a focal adhesion kinase. Cell adhesion generates an essential anti-apoptotic signal (or a viability signal) (Gilmore et al., 2000). Activation of Src family kinases downstream of integrin signaling is common in many cell types (Huveneers et al., 2010 though the outcome of the Src family kinase signaling differs depending upon the cell type and the particular member of the kinase family (src, fyn, yes, fgr, lck, hck blk, lyn and frk). Thus the importance of kinetic constants associated with reactions including these signaling molecules in our model is consistent with existing literature data.

The sensitivity analysis allowed us to examine how changes in two parameters interacted. Horizontal lines in Figures 5.4(a) and 5.4(b) would imply little interaction of the two parameters while diagonal or curved lines imply a significant interaction. From Figures 5.4(a) and 5.4(b), the most significant interaction between parameter appears to be between k_{10} and k_2 and k_3, the rate constant associated with the interaction of Aβ–I* and Src, and the reactions with integrin-ECM and Src. As these represent competing steps in the model, that their rate constants would demonstrate this interdependence is somewhat expected.

Figure 5.5(a): (Sensitivity plot of viability signal values vs. biomolecules at t = 10 hrs. Sensitivity plot depicts main and interaction effects of biomolecules and its effects onto the viability signal. The steepness of the slope determines the magnitude of effect from the parameters. Biomolecules I, Src, pSrc, GAP-GDP and Aβ have the biggest effect and share interdependence. Due to Aβ's presence, viability signal rapidly depletes such that steady state values are effectively 0 μM.)

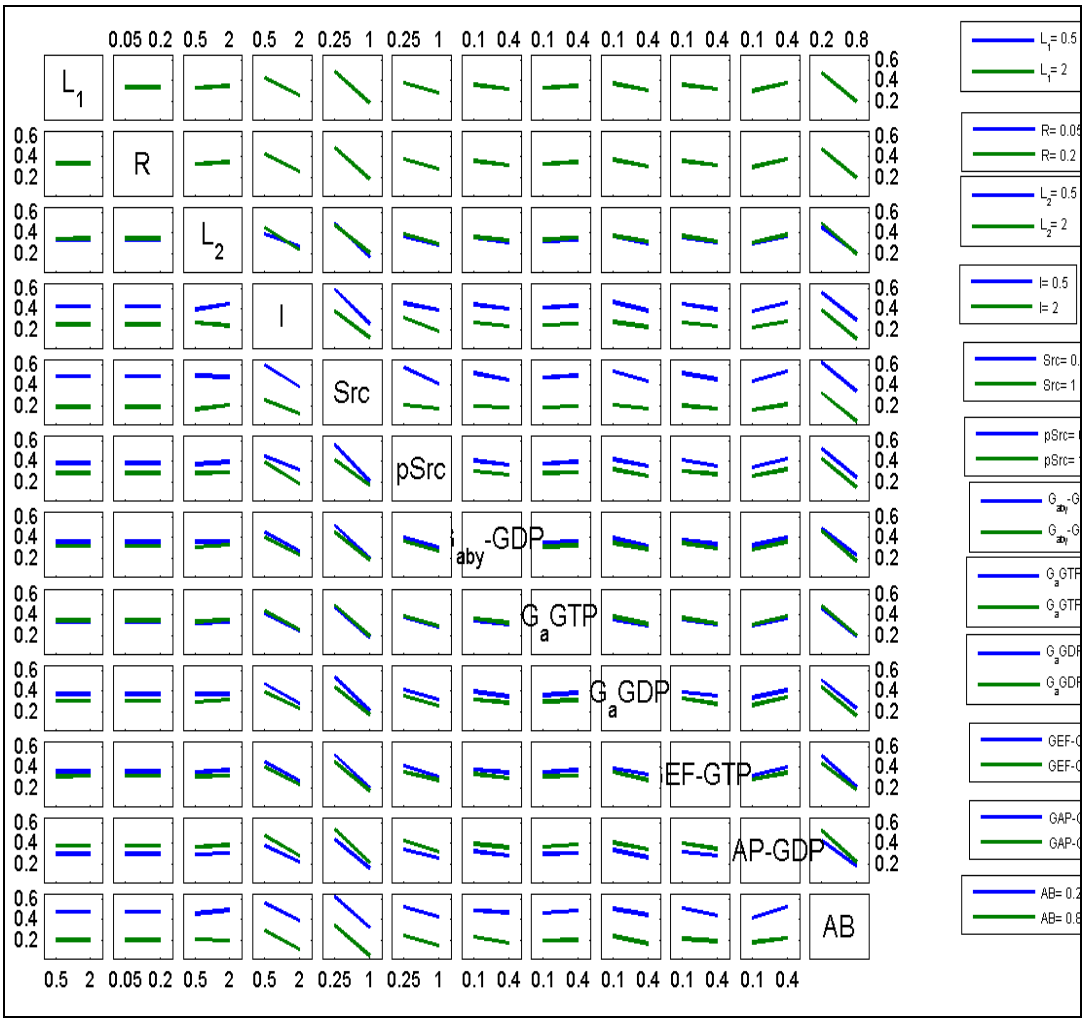
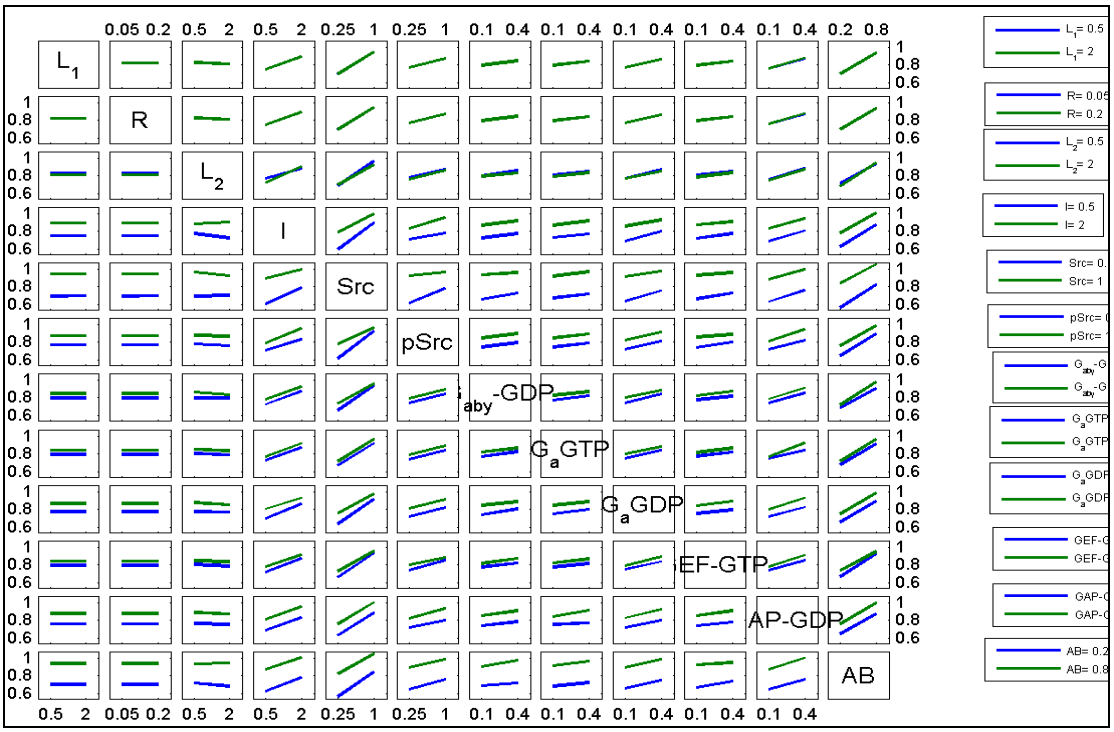


Figure 5.5(b): (Sensitivity plot of death signal values vs. biomolecules at t = 10 hrs. Sensitivity plot depicts main and interaction effects of biomolecules and its effects onto the viability signal. The steepness of the slope determines the magnitude of effect from the parameters. Biomolecules I, Src, pSrc, GAP-GDP and Aβ have the biggest effect and share interdependence. Due to Aβ's presence, death signal rapidly accumulates such that steady state values are effectively 1 μM.)



Figures 5.5(a) and 5.5(b) depict the sensitivity analysis showing the impact of the model initial conditions on the viability and death signals, respectively. In both figures, parameters, integrin (I), Src, pSrc, G-Proteins had the most noticeable effects onto both the viability and death signal while the GDP bound protein solely had an effect onto the death signal. Of course, Aβ affected viability and death signals, with increased Aβ associated with lower viability signals and higher death signals. As concentrations of integrin increase, the viability signal increased and death signal decreased. Under normal conditions, free integrin binds with free ECM (L_2) to

promote cell proliferation. At low levels of integrin, greater levels of ECM (L_2) lead to an increase in the viability signal, consistent with what is known about ECM-integrin interactions and viability, however, at high initial conditions of integrin, ECM initial conditions have little effect on viability or death signals. At low levels of Aβ, high levels of ECM lead to a reduction in the death signal, a trend not seen at high levels of Aβ. We explain these findings as follows: when Aβ is involved, integrin plays a dual role in cell proliferation and neurodegeneration; free integrin is more readily combined with Aβ, which is related to neurodegeneration. Unlike the simplified model, high concentrations of Aβ greatly affect the signaling of Src and pSrc, where higher levels of pSrc and Src are associated with lower levels of viability, as seen in figures 5.5(a) and 5.5(b).

As expected in figure 5.5(a), viability signal increases with GAP, which is related to cell spreading in other systems (Shen et al., 2012). However, in figure 5.5(b), higher values GAP-GDP are related with a higher death signal due to stripping a phosphor group from $G_{\alpha\text{-GTP}}$, converting it into $G_{\alpha\text{-GDP}}$, which reacts with GEF-GTP and increases cell death signal. Additionally, the conversion to $G_{\alpha\text{-GDP}}$ dephosphorylates pSrc, leading to a Src molecule that is free to react with an activated Aβ-integrin complex, Aβ-I*, causing neurodegeneration and increasing the death signal. The model is fairly insensitive to other forms of G protein (GEF,,GaGDP, GaGTP, $G_{\alpha-GDP}G_{\beta\gamma}$), which while not anticipated, is consistent with the relative insensitivity of the model to the initial concentrations of L1 and R, the ligand and G protein coupled receptor, respectively.

Figure 5.6(a): (Sensitivity plot of steady state times vs. kinetic constants. Sensitivity plot depicts main and interaction effects of kinetic constants and its effects onto the steady state time. The steepness of the slope determines the magnitude of effect from the parameters. Kinetic constants k2, k3, k5, k8, k9 and k10 have the most profound effect on time have the biggest effect and share interdependence.)

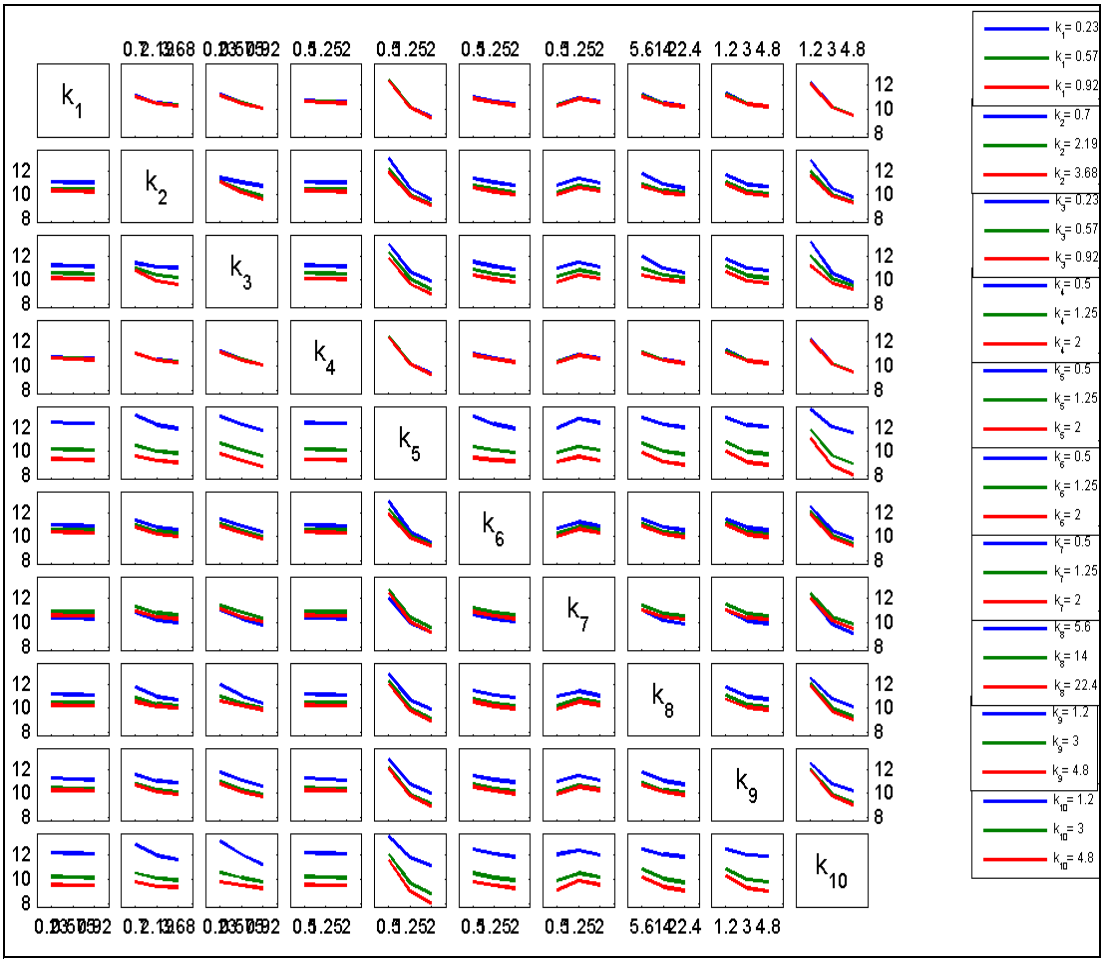
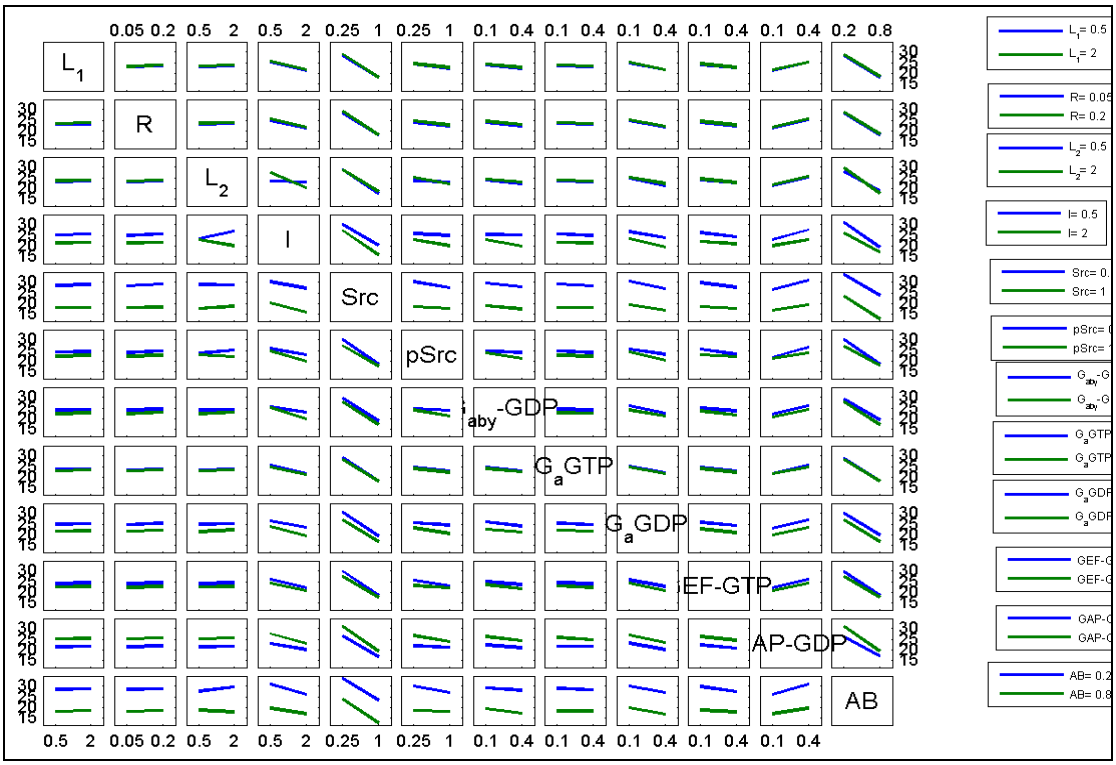


Figure 5.6(b): (Sensitivity plot of steady state times vs. key biomolecules. Sensitivity plot depicts main and interaction effects of the biomolecules and its effects onto the steady state time. The steepness of the slope determines the magnitude of effect from the parameters. Biomolecules, L_2, I, Src, several G-Proteins along with Aβ have the biggest effect.)



Figures 5.6(a) and 5.6(b) depict how varying initial kinetic constants and initial conditions affect the time needed to reach dynamic equilibrium. Generally, steeper slopes among the plots imply higher sensitivity of steady state times with respect to the inputs. In Figure 5.6(a), kinetic constants k_2, k_3, k_5, k_8, k_9 and k_{10} have the most profound effect on time to reach steady state, with k_5 and k_{10} potentially the most impactful. Parameters that display the highest sensitivities are correlated with reactions that have the lowest kinetic constants and comprise of reactants and products that are coupled with other reactions. Such reactions are considered as the rate limiting steps,

which are considered to have the biggest effect onto attaining steady state times for viability and death signals alike.

Similar trends can be seen within Figure 5.6(b) as well. Initial concentrations of Src and Aβ have the biggest impact on equilibrium times. ECM (L_2) increases the amount of time taken to reach equilibrium due to its competition with Aβ. Higher concentrations of Integrin (I) also shortens the amount time needed for equilibrium, but has a lesser effect due to the competition between Aβ and L_2 for free active sites. Finally, GAP-GDP and GEF-GTP increase and decrease the time needed to reach steady state since these are part of the G-protein cycle for cell spreading and retraction, albeit to a lesser extent.

In Figure 5.6(a), it is intuitive to think that these kinetic constants have the biggest effect on the steady state times since these are the reactions that have the slowest reactions times, but also directly impact the viability and death signals. Thus, the higher the rate constant, the quicker the system would reach equilibrium

Model Comparison: Simplified vs G-Protein Model

Within the previous chapters, we have analyzed in depth the simplified and G-Protein model using a combination of stability and sensitivity analysis to understand the dynamics and magnitude of sensitivity between cell survival and death signals, Src signaling, G-protein and integrin interactions. We compared both models to previous work and discarded the Aβ interactions in the simplified model, as those results were not consistent with experimental data. We examined which of the new reactions in

the G- Protein model appear to have the greatest impact on model output, and those most correlated with what we consider to be experimentally consistent predictions. The model output does not appear to be sensitive to the new G protein coupled reactions and G protein effector reactions, but appears very sensitive to the reactions associated with the activated Aβ-integrin complex (reaction 5.17 in particular). Thus, while we and others have observed experimentally that inhibition of GTPase activity somewhere in the Aβ signaling pathway can attenuate Aβ toxicity, it is not clear that we have captured the most salient G-protein signaling within the models.

Chapter 6: Conclusions, Limitations & Future Work

In this thesis, we addressed contrasting theories regarding Aβ's role in AD as reported in literature, previous lab work and other laboratories that have addressed the problem. This led to our hypothesis that the Aβ peptide, in its activated form, disrupts cellular mechanisms linked to learning, memory and cell death, and we proposed that analyzing this interaction and the aberrant cell behavior at the signaling would pinpoint key modes of Aβ interactions that precede cell death. The results also highlight noteworthy pathways, in great detail, of how Aβ's involvement with signaling affect the cell viability signal, which in turn, can be used as a tool to propose future experiments in preventing the disease.

Our work has been classified into two areas, namely, the development of several kinetic models and its stability criteria to describe the behavior of activated Aβ and its effects onto neurons, and sensitivity analysis to explore the impact of kinetic parameters as well as to prove the self-consistency of the models. All the studies reaffirmed that the Aβ peptide in its aggregated form does indeed play a role in AD by altering the cells' natural state especially in the realm of learning, memory and cell survival (Klafki et al., 2006).

We now summarize and present our findings one by one classified by subject, and discuss possible future work that would enhance our current findings and lead to new discoveries in the realm of AD.

Qualitative Trends, Stability & Sensitivity Analysis

Upon interacting with the cellular surface, we hypothesized that the Aβ alters cell homeostasis at the survival level. From the proposed models, we have seen disparate results between the Simplified and G-Protein model. While both models incorporated competitive inhibition from Aβ onto the ligand-integrin interaction, increased concentrations of Aβ, in the Simplified model, failed affecting the cell viability signal due to the lack of interaction with key G-Protein molecules.

In addition to qualitative trends, stability and sensitivity analysis of the Simplified model further reinforced its failure to substantiate claims of complex signaling of Aβ and other key molecules. Within the model, as we have shown, stability and sensitivity analysis was conducted with differing concentrations of Aβ to see its effects onto the viability and death signal. Global asymptotic stability was seen, in the presence and absence of Aβ, from the stability analysis of the Simplified model. However, the sensitivity analysis has shown, regardless of how much Aβ is changed, the viability signal remains insensitive to the change, which is not consistent with previous work (Venkatasubramanian et al., 2014; Rymer et al.). Additionally, increased levels of Aβ have shown to have little effect onto the amount Src phosphorylation seen, which also contradicts previous results of a high correlation between increased levels of Src signaling with increased levels of Aβ. Though the model was simple and intuitive, it did not prove to be self-consistent and was rejected.

Since the Simplified model did not prove to be self-consistent, a more robust mechanism was proposed and developed. Within the G-Protein model, increased values of Aβ have shown to have deleterious effects on the viability signal, as well inducing phosphorylation of Src kinases; affirming that some complex signal integration occurs upon Aβ interaction with cells, leading to viability or neurodegeneration.

Likewise, stability and sensitivity analysis was conducted on this model to prove its self-consistency. From the stability analysis, we have evaluated the system with the presence and absence of Aβ, and like the Simplified model, in both cases, global stability is seen, but with very different results. In the absence of Aβ, viability signal is shown to be 1 with a negligible death signal while viability signal is close to 0, indicating cell death in the presence of Aβ. The sensitivity model further substantiates the stability analysis in illustrating the interdependence of cell viability and death signals with respect to the following molecules: Aβ, Src, pSrc, I and L_2 and GDP/GTP bound G-proteins and their respective reactions. Since k_5 and k_{10} have shown to have the biggest effect onto the signals, we can conclude that there is interdependence between the two reactions due to the inclusion of pSrc and Src within both reactions. The inclusion of these molecules indicate that with increased levels of these molecules, more of the activated form of Aβ, Aβ-I* is formed, leading to an increase in cell death and cell viability signals respectively. Likewise, the increase of L_2 (ECM) mitigates the increase in the death signal since higher concentrations of ECM provide higher potential for the GTP-bound G-protein to form, and react with GAP-GDP, the promoting protein, which is synonymous with a

cell viability signal. Given that the inclusion of Aβ and the interdependence of several key reactions, we have shown self-consistency within the model, and that Aβ has a non-linear effect on cell survival, further affirming that some complex signal integration occurs upon Aβ interaction with cells that can lead to health or neurodegeneration.

These studies led us to deduce possible modes of interaction of the peptide with the integrin receptor and receptors near the cell membrane neurons. The studies also pinpointed multiple modes of binding of Aβ with cells, involving both biological as well as physico-chemical interactions that can lead to neurodegeneration. The studies further highlighted the role of Src kinase and its role in the presence and absence of Aβ, providing understanding for future researchers in preventing the toxicity of the Aβ peptide.

Limitations

With our proposed models, detailed kinetic models have contributed significantly to our understanding of the principles of metabolic regulation, providing quantitative and qualitative findings to the crosstalk of biomolecules. However, despite their general applicability, the construction of large kinetic models faces a number of significant difficulties.

In contrast to chemical systems, kinetic parameters in biological systems are often context specific. For example, the catalytic activity of enzymes may depend on multiple factors such as temperature, pH, and multiple other conditions in a nonlinear fashion. As a result, it becomes exceedingly difficult to obtain reliable estimates of

kinetic parameters, forcing us to use literature values with wide ranges, depending on the procedure and operating conditions to obtain the constants.

Additionally, literature values were not readily or easily attainable for a number of biochemical interactions for both mechanisms. Thus, educated estimates, for both unknown forward and reversible rates, were scaled with respect to known kinetic constants to have a working model. Given the general uncertainty of the kinetic constants, concentration profiles, stability and sensitivity analyses could significantly change.

Furthermore, both mechanisms solely focused onto the kinetic aspects of Aβ aggregation and its effects on the G-Protein system. As mentioned in **Chapter 1**, crosstalk interactions among biomolecules are intra- and intercellular, indicating possible mass transfer limitations, which are not taken into account in either model. Rate limiting steps would have to be experimentally determined to obtain accurate scaling of kinetic constants, and mass transfer equations would have to be numerically calculated alongside kinetic expressions, forcing the use of a partial differential equation (PDE) solver, further complicating the model.

Finally, detailed kinetic models are computationally intensive, even for a relatively small number of reactions. Computationally intensive models limit the scope of the parameter space for sensitivity analysis, perpetuating oversight of possible regions of interest when compared to modeling schemes that allow more flexibility.

Future Directions

The role of Aβ-induced signaling pathways can be investigated further to generate a more complete picture of the pathways that it affects. However, as previously mentioned, detailed kinetic modeling will prove to be too cumbersome to handle the increase in the number of parameters and variables evaluate (Steuer, 2009). Hence, a combination of structured kinetic modeling with elements of stochastic modeling can be incorporated to further evaluate the parameter space, and non-linear stability analysis can be used in conjunction to answer the stability criterion more completely (Steuer, 2009). Understanding the role of the different parameters in determining the stability of the system, and how changes in parameters would lead to differences in stability could point to sensitive areas of the signaling network in determining cell fate, or for exploring therapeutic interventions. Finally, a sum of the square residuals can be calculated to select a set of kinetic parameters to more accurately capture experimental trends (Steuer, 2009).

Concluding Summary

This thesis highlighted the crucial aspects of Aβ's role in cellular mechanisms, illustrated key points of Aβ's interaction with the cell membrane and its mediated alteration of signaling pathways, and provided a foundation for extensive future research into using systems engineering tools (stability and sensitivity analysis) to qualitatively trend, characterize and predict key behavior changes of in cells upon exposure to Aβ. The system engineering tools also help us quickly isolate which additional reactions contributed to the improved model predictions. Finally, the

analysis enables us to relate model parameters with experimental findings in a manner that could help us to identify future experiments or therapeutic avenues to change the outcome of the signaling network, and more specifically, to alter neurotoxicity associated with Aβ.

Appendices

FullFactorial_FullModelAB

```
function fullfactorial_FullModelAB
    % for initial conditions variable
    numlevels = 2; % number of levels for each variable
    numEqns = 20;
   FullFactorialKCs(numlevels,numEqns);
   %FullFactorialICs(numlevels,numEqns);
end

function FullFactorialKCs(numlevels,numEqns)
    numparam = 10; % number of variables changing for DoE
    ff = numlevels*ones(1,numparam);
    kdff = fullfact(ff);
    numRuns = numlevels^numparam;
    %Grouping variables with numlevels number of levels
    kVar = zeros(numparam,numlevels);

    kC=zeros(numRuns,numparam);
    k1low = .23;   k1hi = 4*k1low;
    k2low = .7;    k2hi = 4*k1hi;
    k3low = .23;   k3hi = 4*k3low;
    k4low = .50;   k4hi = 4*k4low;    %Aru thesis
    k5low = .50;   k5hi = 4*k5low;
    k6low = .50;   k6hi = 4*k6low;
    k7low = .50;   k7hi = 4*k7low;
    k8low = 5.6;   k8hi = 4*k8low;
    k9low = 1.2;   k9hi = 4*k9low;    %Aru thesis
    k10low = 1.2;  k10hi = 4*k10low;  %Aru thesis

    kVar(1,:) = linspace(k1low,k1hi,numlevels);
    kVar(2,:) = linspace(k2low,k2hi,numlevels);
    kVar(3,:) = linspace(k3low,k3hi,numlevels);
    kVar(4,:) = linspace(k4low,k4hi,numlevels);
    kVar(5,:) = linspace(k5low,k5hi,numlevels);
    kVar(6,:) = linspace(k6low,k6hi,numlevels);
    kVar(7,:) = linspace(k7low,k7hi,numlevels);
    kVar(8,:) = linspace(k8low,k8hi,numlevels);
    kVar(9,:) = linspace(k9low,k9hi,numlevels);
    kVar(10,:) = linspace(k10low,k10hi,numlevels);

    % Set value of n for levels in factorial
    for i = 1:numRuns
        for j = 1:numparam
            if kdff(i,j) == 1
                kC(i,j) = kVar(j,1);
            elseif kdff(i,j) == 2
                kC(i,j) = kVar(j,2);
            elseif kdff(i,j) == 3
                kC(i,j) = kVar(j,3);
            elseif kdff(i,j) == 4
```

```
                kC(i,j) = kVar(j,4);
            elseif kdff(i,j) == 5
                kC(i,j) = kVar(j,5);
            end
        end
    end
    kC = kC'; % transposing matrix

    numConst = 7;
    kConst = zeros(numConst,numRuns);
    kConst(1,:) = .25;      %k1r
    kConst(2,:) = .25;      %k2r
    kConst(3,:) = .25;      %k3r
    kConst(4,:) = .25;      %k4r
    kConst(5,:) = .1;       %k5r
    kConst(6,:) = .1;       %k8r
    kConst(7,:) = .1;       %k9r

    %%Initial Conditions
    %Modify ligand/receptors and integrin
    %Only for the first array within IC.
    %IC(1):L1                IC(2):R                     IC(3):L1R*
    %IC(4):L2                IC(5):I                     IC(6):L2I*
    %IC(7):SRC               IC(8):pSRC                  IC(9):Ga-GDP-
Gby
    %IC(10):Ga-GTP           IC(11):Gby                  IC(12):Ga-GDP
    %IC(13):GEF-GTP          IC(14):V                    IC(15):GEF-GDP
    %IC(16):D                IC(17):GAP-GDP              IC(18):GAP-GTP
    %IC(19):AB               IC(20):AB-I*

    iC=zeros(numEqns,numRuns);
    %iC(1,2,4,5,7,8,13) = uM
    %iC(1,2,4,5) = (1,.1,1,1)
    iC(1,:) = .5;    iC(2,:) = .05;   iC(3,:) = 0;      iC(4,:) =
.5;
    iC(5,:) = .5;    iC(6,:) = 0;     iC(7,:) = .25;    iC(8,:) =
.25;
    iC(9,:) = .1;    iC(10,:) = .1;   iC(11,:) = 0;     iC(12,:) =
.1;
    iC(13,:) = .1;   iC(14,:) = 1;    iC(15,:) = 0;     iC(16,:) =
0;
    iC(17,:) = .1;   iC(18,:) = 0;    iC(19,:) = 0;    iC(20,:) = 0;
    %iC(19,:) = 0.5;
    IC = iC;

    save('narf_iC','iC')
    save('narfIC','IC')
    save('narf_kC','kC')
    save('narf_kConst','kConst')
    save('narf_numruns','numRuns')
end

function FullFactorialICs(numlevels,numEqns)

    numparam = 12; % number of variables changing for DoE
    ff = numlevels*ones(1,numparam);
```

```
    dff = fullfact(ff); %length(dff) = total number of runs
    numRuns = numlevels^numparam;

    %for initial conditions variable
    %Initial Conditions
    %Only for the first array within IC.
    %IC(1):L1              IC(2):R                  IC(3):L1R*(0)
    %IC(4):L2              IC(5):I                  IC(6):L2I* (0)
    %IC(7):SRC             IC(8):pSRC               IC(9):Ga-GDP-
Gby
    %IC(10):Ga-GTP         IC(11):Gby(0)            IC(12):Ga-GDP
    %IC(13):GEF-GTP        IC(14):V                 IC(15):GEF-GDP
(No IC)
    %IC(16):D(0)           IC(17):GAP-GDP           IC(18):GAP-GTP
(No IC)
    %IC(19):AB             IC(20):AB-I*(0)

    IC1low = .50;   IC1hi = 4*IC1low;
    IC2low = .05;   IC2hi = 4*IC2low;
    IC4low = .50;   IC4hi = 4*IC4low;    %Aru thesis
    IC5low = .50;   IC5hi = 4*IC5low;
    IC7low = .25;   IC7hi = 4*IC7low;
    IC8low = .25;   IC8hi = 4*IC8low;
    IC9low = .10;   IC9hi = 4*IC9low;
    IC10low = .10;  IC10hi = 4*IC10low;  %Aru thesis
    IC12low = .10;  IC12hi = 4*IC12low;  %Aru thesis
    IC13low = .10;  IC13hi = 4*IC13low;  %Aru thesis
    IC17low = .10;  IC17hi = 4*IC17low;  %Aru thesis
    IC19low = .20;  IC19hi = 4*IC19low;  %Aru thesis
    %IC19low = 0;   IC19hi = 4*IC19low;  %Aru thesis

    IVar = zeros(numparam,numlevels);

    IVar(1,:) = linspace(IC1low,IC1hi,numlevels);
    IVar(2,:) = linspace(IC2low,IC2hi,numlevels);
    IVar(3,:) = linspace(IC4low,IC4hi,numlevels);
    IVar(4,:) = linspace(IC5low,IC5hi,numlevels);
    IVar(5,:) = linspace(IC7low,IC7hi,numlevels);
    IVar(6,:) = linspace(IC8low,IC8hi,numlevels);
    IVar(7,:) = linspace(IC9low,IC9hi,numlevels);
    IVar(8,:) = linspace(IC10low,IC10hi,numlevels);
    IVar(9,:) = linspace(IC12low,IC12hi,numlevels);
    IVar(10,:) = linspace(IC13low,IC13hi,numlevels);
    IVar(11,:) = linspace(IC17low,IC17hi,numlevels);
    IVar(12,:) = linspace(IC19low,IC19hi,numlevels);
    save('narf_var','IVar')

    % Set value of n for levels in factorial
    for i = 1:numRuns
        for j = 1:numparam
            if dff(i,j) == 1
                iC(i,j) = IVar(j,1);
            elseif dff(i,j) == 2
                iC(i,j) = IVar(j,2);
            elseif dff(i,j) == 3
                iC(i,j) = IVar(j,3);
```

```
            elseif dff(i,j) == 4
                iC(i,j) = IVar(j,4);
            elseif dff(i,j) == 5
                iC(i,j) = IVar(j,5);
            end
        end
    end
    iC = iC'; %transposed matrix
    save('narf_iC','iC')

    IC = zeros(numEqns,numRuns); % Set of Initial Condition matrix
    %IC(3,6,11,16,20) = 0;
    IC(1,:) = iC(1,:);      IC(2,:) = iC(2,:);     IC(4,:) = iC(3,:);
    IC(5,:) = iC(4,:);      IC(7,:) = iC(5,:);     IC(8,:) = iC(6,:);
    IC(9,:) = iC(7,:);      IC(10,:) = iC(8,:);    IC(12,:) = iC(9,:);
    IC(13,:) = iC(10,:);    IC(17,:) = iC(11,:);   IC(19,:) =
iC(12,:);
    IC(14,:) = 1;           IC(15,:) = .1;         IC(18,:) = .1;


    kC=zeros(10,numRuns);
    k1low = .23;    %k1hi = 4*k1low;
    k2low = .7;     %k2hi = 4*k1hi;
    k3low = .23;    %k3hi = 4*k3low;
    k4low = .50;    %k4hi = 4*k4low;     %Aru thesis
    k5low = .50;    %k5hi = 4*k5low;
    k6low = .50;    %k6hi = 4*k6low;
    k7low = .50;    %k7hi = 4*k7low;
    k8low = 5.6;    %k8hi = 4*k8low;
    k9low = 1.21;   %k9hi = 4*k9low;     %Aru thesis
    k10low = 1.21; %k10hi = 4*k10low;   %Aru thesis

    kC(1,:) = k1low;     kC(2,:) = k2low;     kC(3,:) = k3low;
    kC(4,:) = k4low;     kC(5,:) = k5low;     kC(6,:) = k6low;
    kC(7,:) = k7low;     kC(8,:) = k8low;     kC(9,:) = k9low;
    kC(10,:) = k10low;

    kConst = zeros(7,1);
    kConst(1,:) = .25;      %k1r
    kConst(2,:) = .25;      %k2r
    kConst(3,:) = .25;      %k3r
    kConst(4,:) = .25;      %k4r
    kConst(5,:) = .1;       %k5r
    kConst(6,:) = .1;       %k8r
    kConst(7,:) = .1;       %k9r

    save('narf_iC','iC')
    save('narfIC','IC')
    save('narf_kC','kC')
    save('narf_kConst','kConst')
    save('narf_numruns','numRuns')
end
```

FullModelABFact_151210

```
function FullModelABfact_151210
%%Preliminary
Setup%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% number of equations to solve for.
global numEqns numRuns;
numEqns = 20;
load('narf_iC','iC')
load('narfIC','IC')
load('narf_kC','kC')
load('narf_kConst','kConst')
load('narf_numruns','numRuns') %number of set of ICs for respective
soln.
k = kC; % Rate Constants
%%Multiple ODE Solution
Solver%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t_initial = 0;
t_sens = 10; % for sensitivity plots
t_final = 200;
tspan = [t_initial,t_final];
options = odeset('RelTol',1e-6,'AbsTol',1e-6);
SSValues_V = zeros(numRuns,1);
SSValues_D = SSValues_V;
SSTime = zeros(numRuns,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%Loops for through set of Initial Conditions to obtain soln
(numSolns)
for i = 1:numRuns
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [t,y] = ode23s(@(t,y) FullModelFunc(t,y,k(:,i),kConst),...
                   tspan,IC(:,i),options);
    A = [t,y];  % matrix of interest for all variables of interest
    %save('ODERuns','A','-ascii','-append')
    PlotODE(i,t,y,k,IC,numEqns);
    %PutODEIntoExcel(i,A,IC,numRuns);
    sensi_array = find(t>=t_sens);
    sensi_elem = sensi_array(1);
    SSValues_V(i) = y(sensi_elem,14); %last element of V := steady
state value
    SSValues_D(i) = y(sensi_elem,16); %last element of D := steady
state value
    SSTimePoint = FindSStime_ODE(i,t,y);
    SSTime(i) = t(SSTimePoint);
    display(i);
end
k = k'; % Match up dimensions
iC = iC';
PlotInteractions(SSValues_V,SSValues_D,SSTime,k,iC);
function dydt = FullModelFunc(~,y,k,kConst)
    %Rate Laws%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Previous Values for calculating next iteration
    %y(1):L1              y(2):R                    y(3):L1R*
    %y(4):L2              y(5):I                    y(6):L2I*
    %y(7):SRC             y(8):pSRC                 y(9):Ga-GDP-Gby
    %y(10):Ga-GTP         y(11):Gby                 y(12):Ga-GDP
```

```
    %y(13):GEF-GTP        y(14):V                 y(15):GEF-GDP
    %y(16):D              y(17):GAP-GDP           y(18):GAP-GTP
    %y(19):AB             y(20):AB-I*

    k1r = kConst(1);    k2r = kConst(2);    k3r = kConst(3);
    k4r = kConst(4);    k5r = kConst(5);    k8r = kConst(6);
    k9r = kConst(7);

    r1 = -k(1)*y(1)*y(2) + k1r*y(3); %check!
    r2 = -k(2)*y(4)*y(5) + k2r*y(6); %check!
    r3 = -k(3)*y(6)*y(7) + k3r*y(4)*y(5)*y(8); %check!
    r4 = -k(4)*y(9)*y(3) + k4r*y(1)*y(2)*y(10)*y(11); %check!
    r5 = -k(5)*y(8)*y(12)+ k5r*y(7)*y(10); %check!

    r6 = -k(6)*y(12)*y(13)*y(14); %check!
    r7 = -k(7)*y(10)*y(17)*y(14); %check!
    r8 = -k(8)*y(19)*y(5) + k8r*y(20); %check!
    r9 = -k(9)*y(20)*y(9) + k9r*y(19)*y(5)*y(12)*y(11);
    r10 = -k(10)*y(20)*y(7)*y(14); %check

    %ODES%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % dydt represents an ode solved for an individual variable
    dydt=zeros(numEqns,1);
    dydt(1) = r1 - r4; %L1 check!
    dydt(2) = r1 - r4; %R check!
    dydt(3) = -(r1 - r4); %L1R* check!
    dydt(4) = r2 - r3; %L2 check!
    dydt(5) = r2 - r3 + r8 - r9 - r10; %I check!
    dydt(6) = -(r2 - r3); %L2I* check!
    dydt(7) = r3 - r5 + r10;  %Src check!
    dydt(8) = -(r3 - r5 + r10); %pSrc check!
    dydt(9) = r4 + r9;  %GaGDPGby check!
    dydt(10) = -r4 - r5 - r6 + r7; %GaGTP check!

    dydt(11) = -r4 - r9; %Gby check!
    dydt(12) = r5 + r6 - r7 - r9; %GaGDP check!

    dydt(13) = r6; %GEF-GTP check!
    dydt(14) = r6 - r7 + r10; %V check!
    dydt(15) = -r6; %GEF-GDP check!

    dydt(16) = -r6 - r10; %D check!
    dydt(17) = r7; %GAP-GDP check!
    dydt(18) = -r7; %GAP-GTP check!
    dydt(19) = r8 - r9 - r10; %AB check!
    dydt(20) = -(r8 - r9 - r10); %AB-I* check!
end

function PutODEIntoExcel(i,A,IC,numRuns)
    %For excel file matrix
    %Adds a "time" element to ICset array (set @ 0)
    %ICset' is transposed to match dimensions of IC
    % To prepare the header for the excel file.
    ICtimefill = zeros(1,numRuns);
    ICset = [ICtimefill;IC]';
```

```
    header = {'t','L1','R','L1-R*','L2','I','L2-
I*','Src','pSrc','Ga-GDP-Gby',...
          'Ga-GDP','Gby','Ga-GTP','GEF-GTP','V','GEF-GDP','D','GAP-
GDP',...
          'GAP-GTP','AB','AB-I*'};

    %xlswrite file of data of interest
     Solnmat = [ICset(i,:);A]; %Solnmat = Solution Matrix
     xlswrite('1',Solnmat,i);
     xlswrite('1',header,i);
end


function SSTimePoint = FindSStime_ODE(i,t,y)


%https://www.mathworks.com/help/matlab/matlab_prog/vectorization.htm
l
    %ICtimefill = zeros(1,numRuns);
    %ICset = [ICtimefill;IC]';
    %header = {'t','L1','R','L1-R*','L2','I','L2-
I*','Src','pSrc','AB',...
    %          'AB-I*','D','X*','V'};
    %header1 = {'dL1','dR','dL1-R*','dL2','dI','dL2-
I*','dSrc','dpSrc','dAB',...
    %          'dAB-I*','dD','dX*','dV'};
    tol = .01;
    diffA = diffxy(t,y);

    L = logical(diffA < tol);
    B = all(L == 1, 2);
    indexarray = find(B);
    SSTimePoint = indexarray(1); %Finds first occurence of reaching
SS
end
end
```

PlotODE

```
function PlotODE(i,t,y,k,IC,numEqns)
    %%Color Scheme for plotting
    cb1 = [255 0 255];   cb2 =  [0 0 0];     cb3 =  [0 255 255];
    cr1 = [255 0 0];     cr2 = [0 0 255];    cr3 = [0 255 0];
    cgr1 = [11 141 255]; cgr2 =  [155 3 31]; cgr3 = [243 91 57];
    c1 = [253 10 120];   c2 = [156 168 26];  c3 = [178 138 117];
    cc = [cb1;cr1;cgr1;c1;cb2;cr2;cgr2;c2;cb3;cr3;cgr3;c3]./255;

    plotsymbol = {'-','--','-.'};  %array of plotting symbols

    %%legend string array
    legendstr = {'L1','R','L2','I','Src','pSrc','G\alpha-GTP',...
                 'G\alpha-GDP','V','D','A\beta'};

    phasestr = {'L1','R','L2','I','Src','pSrc','G\alpha-GTP',...
                 'G\alpha-GDP','A\beta'};

    PlotODE_ICs(i,t,y,cc,IC,legendstr,numEqns);
    %PlotODE_kCs(i,y,k,cc,legendstr,numEqns)
    %PlotODE_ICs_VD(i,y)
    %PlotODE_kCs_VD(i,y)
    PhasePlots(i,y,cc,plotsymbol,phasestr);
function PlotODE_ICs(i,t,y,cc,IC,legendstr,numEqns)
    %%Plot Color & Symbol Scheme
    PoNI = 9; %plots of no interest
    plotsymbol = {'-','--','-.'};  %array of plotting symbols
    numplots = numEqns-PoNI;  %11 plots of interest.

    isym = 1:numplots; %to ensure repeating color/symbol combo
doesn't occur.
    %sorts element 1,1,2,2,3,etc...
    symbol = sort(mod(isym,length(plotsymbol))+1);
    %unsorted element: 1,2,3,4,5,6,7,8,9,1,2,etc...
    col = mod(isym,length(cc))+1;

    %%Plotting%%%%%%%%%%%%%%%%%%%%%%%%
    %IC(1):L1              IC(2):R                 IC(3):L1R*
    %IC(4):L2              IC(5):I                 IC(6):L2I*
    %IC(7):SRC             IC(8):pSRC              IC(9):Ga-GDP-
Gby
    %IC(10):Ga-GTP         IC(11):Gby              IC(12):Ga-GDP
    %IC(13):GEF-GTP        IC(14):V                IC(15):GEF-GDP
    %IC(16):D              IC(17):GAP-GDP          IC(18):GAP-GTP
    %IC(19):AB             IC(20):AB-I*
    %Plotting array for all variables of interest
     plotarray = zeros(length(t),numplots);
     plotarray(:,1) = y(:,1);     plotarray(:,7) = y(:,10);
     plotarray(:,2) = y(:,2);     plotarray(:,8) = y(:,12);
     plotarray(:,3) = y(:,4);     plotarray(:,9) = y(:,14);
     plotarray(:,4) = y(:,5);     plotarray(:,10) = y(:,16);
     plotarray(:,5) = y(:,7);     plotarray(:,11) = y(:,19);
     plotarray(:,6) = y(:,8);
```

```
   %  n = 1:numplots;
   %      plotarray(:,n) = y(:,n);
     set(gcf,'Visible','off')              % turns current figure
"off"
     set(0,'DefaultFigureVisible','off');  % all subsequent figures
"off"
     f = figure;
     for j = 1:numplots;
         plot(t,plotarray(:,j),plotsymbol{symbol(j)},...
         'color',cc(col(j),:),'LineWidth',1.5);
         hold on
     end
     %axis([0 200 0 1.1])
     %ax = gca;
     %set(ax,'YTick',[0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0 1.1])

     ylab = ylabel('Metabolite Concentrations ( \muM)');
     titlab = title('Metabolite Concetrations vs. Time');
     leglab = legend(legendstr);

      %%%%%%Setting up time label for plot
      label_time = sprintf('Time (hr) \n');
      str1 = sprintf(...
      'L_{10} = %g, R_0 = %g, L_{20} = %g, I_0 = %g',...
      IC(1,i),IC(2,i),IC(4,i),IC(5,i));
      str2 = sprintf(...
      'Src_0= %g, pSrc_0= %g , G\\alpha-GTP_0 = %g, G\\alpha-GDP_0 =
%g',...
      IC(7,i),IC(8,i),IC(10,i),IC(12,i));
      str3 = sprintf(...
      'GEF-GTP_0= %g, GAP-GDP_0 = %g, A\\beta_0 = %g, V_0 = %g',...
      IC(13,i),IC(17,i),IC(19,i),IC(14,i));
      str4 = sprintf('G\\alpha-GDPG\\beta\\gamma_0= %g',IC(9,i));
      format compact
      xlab = xlabel({label_time;str1;str2;str3;str4});
      format compact
      %%%%end comment

      set(ylab,'FontSize',18);
      set(titlab,'FontSize',18);
      set(leglab,'location','bestoutside','FontSize',14);
      set(xlab,'FontSize',16);
      format compact
      print(f,'-dpng', num2str(i));  % Prints graph to png file, i
as fn.
end
function PlotODE_kCs(i,y,k,cc,legendstr,numEqns)
    %%Plot Color & Symbol Scheme
    PoNI = 2; %plots of no interest
    plotsymbol = {'-','--','-.'};  %array of plotting symbols
    numplots = numEqns-PoNI;  %11 plots of interest.

    isym = 1:numplots; %to ensure repeating color/symbol combo
doesn't occur.
    %sorts element 1,1,2,2,3,etc...
```

```
    symbol = sort(mod(isym,length(plotsymbol))+1);
    %unsorted element: 1,2,3,4,5,6,7,8,9,1,2,etc...
    col = mod(isym,length(cc))+1;

     %Plotting array for all variables of interest
     plotarray = zeros(length(t),numplots);
     plotarray(:,1) = y(:,1);     plotarray(:,8) = y(:,10);
     plotarray(:,2) = y(:,1);     plotarray(:,9) = y(:,12);
     plotarray(:,3) = y(:,4);     plotarray(:,10) = y(:,13);
     plotarray(:,4) = y(:,5);     plotarray(:,11) = y(:,14);
     plotarray(:,5) = y(:,7);     plotarray(:,12) = y(:,16);
     plotarray(:,6) = y(:,8);     plotarray(:,13) = y(:,17);
     plotarray(:,7) = y(:,9);     plotarray(:,14) = y(:,19);

    %plotarray = zeros(length(t),numplots);
    % n = 1:10;
    %     plotarray(:,n) = y(:,n);
    % plotarray(:,11) = y(:,12);
     set(gcf,'Visible','off')              % turns current figure
"off"
     set(0,'DefaultFigureVisible','off');  % all subsequent figures
"off"
     f = figure;
     for j = 1:numplots;
         plot(t,plotarray(:,j),plotsymbol{symbol(j)},...
         'color',cc(col(j),:),'LineWidth',1.5);
         hold on
      end
      axis tight
      ylab = ylabel('Metabolite Concentrations');
      titlab = title('Metabolite Concetrations vs. Time');
      leglab = legend(legendstr);
      %%%%%%Setting up time label for plot
      label_time = sprintf('Time \n');
      str1 = sprintf(...
      'k_1= %g, k_2 = %g, k_3 = %g, k_4 = %g, k_5 = %g',...
      k(1,i),k(2,i),k(3,i),k(4,i),k(5,i));
      str2 = sprintf(...
      'k_6= %g, k_7 = %g, k_8 = %g, k_9 =
%g',k(6,i),k(7,i),k(8,i),k(9,i));
      xlab = xlabel({label_time;str1;str2});
      %%%%end comment

      set(ylab,'FontSize',18);
      set(titlab,'FontSize',18);
      set(leglab,'location','bestoutside','FontSize',18);
      set(xlab,'FontSize',15);
      format compact
      print(f,'-dpng', num2str(i));  % Prints graph to png file, i
as fn.


      g = figure;
      plot(t,y(:,11),t,y(:,13),'LineWidth',1.5)
      xlabel('Time')
      ylabel('Metabolite Concentrations')
```

```
      title('Metabolite Concentrations vs Time')
      legend('D','V');
      print(g,'-dpng', num2str(512+i));  % Prints graph to png file,
i as fn.

end
function PlotODE_ICs_VD(i,y)
      set(gcf,'Visible','off')              % turns current figure
"off"
      set(0,'DefaultFigureVisible','off');  % all subsequent figures
"off"
      g = figure;
      label_time = sprintf('Time \n');
      plot(t,y(:,13),t,y(:,11),t,y(:,7),t,y(:,8),'LineWidth',1.5);
      xmin = 0; xmax = 15; ymin = 0; ymax = 1.4;
      axis([xmin xmax ymin ymax])
      str1 = sprintf(...
      'L_{10}= %g, R_0 = %g, L_{20} = %g, I_0 = %g, Src_0 = %g,
pSrc_0 = %g',...
      IC(1,i),IC(2,i),IC(4,i),IC(5,i),IC(7,i),IC(8,i));
      str2 = sprintf(...
      'G_{aby}-GDP_0 = %g, G_aGTP_0 = %g, G_aGDP_0 = %g',...
      IC(9,i),IC(10,i),IC(12,i));
      str3 = sprintf(...
      'GEF-GTP_0 = %g, GAP-GDP_0 = %g, AB_0 = %g',
IC(13,i),IC(17,i),IC(19,i));

      xlab = xlabel({label_time;str1;str2;str3});
      y_label = sprintf('Metabolite Concentrations');
      leglab = legend('V','D','Src','pSrc');
      titlab = title('Metabolite Concetrations vs. Time');
      xlab = xlabel({label_time;str1;str2;str3});
      ylab = ylabel({y_label});

      set(xlab,'FontSize',14);
      set(ylab,'FontSize',18);
      set(leglab,'FontSize',18)
      set(titlab,'FontSize',18);
      print(g,'-dpng', num2str(i));  % Prints graph to png file, i
as fn.
end
function PlotODE_kCs_VD(i,y)
      set(gcf,'Visible','off')              % turns current figure
"off"
      set(0,'DefaultFigureVisible','off');  % all subsequent figures
"off"
      g = figure;
      label_time = sprintf('Time \n');
      plot(t,y(:,13),t,y(:,11),t,y(:,7),t,y(:,8),'LineWidth',1.5);
      xmin = 0; xmax = 15; ymin = 0; ymax = 1.4;
      axis([xmin xmax ymin ymax])
      str1 = sprintf(...
      'k_1= %g, k_2 = %g, k_3 = %g, k_4 = %g, k_5 = %g',...
      k(1,i),k(2,i),k(3,i),k(4,i),k(5,i));
      str2 = sprintf(...
```

```
      'k_6= %g, k_7 = %g, k_8 = %g, k_9 =
%g',k(6,i),k(7,i),k(8,i),k(9,i));
      xlab = xlabel({label_time;str1;str2});
      set(xlab,'FontSize',15);
      ylabel('Metabolite Concentrations')
      title('Metabolite Concentrations  vs Time')
      legend('V','D','Src','pSrc')
      print(g,'-dpng', num2str(i));  % Prints graph to png file, i
as fn.
end
function PhasePlots(i,y,cc,plotsymbol,phasestr)
    numplots = 9; %plots of interest
    plotarray = zeros(length(t),numplots);
    k = 1:numplots; %to ensure repeating color/symbol combo doesn't
occur.
    symbol = sort(mod(k,length(plotsymbol))+1); %sorts element
1,1,2,2,3,etc...
    col = mod(k,length(cc))+1; %unsorted element:
1,2,3,4,5,6,7,8,9,1,2,etc...

    plotarray(:,1) = y(:,1);     plotarray(:,7) = y(:,10);
    plotarray(:,2) = y(:,2);     plotarray(:,8) = y(:,12);
    plotarray(:,3) = y(:,4);     plotarray(:,9) = y(:,19);
    plotarray(:,4) = y(:,5);
    plotarray(:,5) = y(:,7);     plotarray(:,10) = y(:,14); %V
    plotarray(:,6) = y(:,8);

    ind = find(plotarray(:,10)== max(plotarray(:,10)));

    set(gcf,'Visible','off')              % turns current figure
"off"
    set(0,'DefaultFigureVisible','off');  % all subsequent figures
"off"

    g = figure;
    for j = 1:numplots;

plot(plotarray(:,j),plotarray(:,10),plotsymbol{symbol(j)},...
       'color',cc(col(j),:),'LineWidth',1.10);
        hold on
    end
    legend(phasestr,'location','bestoutside','FontSize',16);
    for j = 1:numplots;
       plot(plotarray(1,j),plotarray(1,10),'o','LineWidth',...

1.25,'MarkerEdgeColor','k','MarkerFaceColor',cc(col(2),:),...
             'MarkerSize',12);
       plot(plotarray(ind,j),plotarray(ind,10),'s','LineWidth',...

1.25,'MarkerEdgeColor','k','MarkerFaceColor',cc(col(1),:),...
             'MarkerSize',12);
       plot(plotarray(end,j),plotarray(end,10),'s','LineWidth',...

1.25,'MarkerEdgeColor','k','MarkerFaceColor',cc(col(7),:),...
             'MarkerSize',12);
    end
```

```
    %axis([0 .50 0 1.01])
    %ax = gca;
    %set(ax,'YTick',[0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0 1.1])
    %axis 'auto x'
    %axis image

    ylab1 = ylabel('Viability');
    set(ylab1,'FontSize',16);
    label_conc = sprintf('Metabolite Concentrations \n');
    str1 = sprintf(...
   'L_{10} = %g, R_0 = %g, L_{20} = %g, I_0 = %g',...
    IC(1,i),IC(2,i),IC(4,i),IC(5,i));
    str2 = sprintf(...
   'Src_0= %g, pSrc_0= %g , G\\alpha-GTP_0 = %g, G\\alpha-GDP_0 =
%g',...
    IC(7,i),IC(8,i),IC(10,i),IC(12,i));
    str3 = sprintf(...
   'GEF-GTP_0= %g, GAP-GDP_0 = %g, A\\beta_0 = %g, V_0 = %g',...
    IC(13,i),IC(17,i),IC(19,i),IC(14,i));
    str4 = sprintf('G\\alpha-GDPG\\beta\\gamma_0= %g',IC(9,i));

    format compact
    xlab1 = xlabel({label_conc;str1;str2;str3;str4});format compact
    set(xlab1,'FontSize',16);
    titlab1 = title('Viability vs. Metabolite Concetrations (G-
Protein w/o A\beta)');
    set(titlab1,'FontSize',18);
    print(g,'-dpng', num2str(i+100000));

end
end
```

PlotInteractions

```
function PlotInteractions(SSValues_V,SSValues_D,SSTime,k,iC)
    PlotInteractions_kCs(SSValues_V,SSValues_D,SSTime,k);
    %PlotInteractions_ICs(SSValues_V,SSValues_D,SSTime,iC);
end
 function PlotInteractions_kCs(SSValues_V,SSValues_D,SSTime,k)
    interactionstr = {'k_1','k_2','k_3','k_4','k_5','k_6','k_7',...
                      'k_8','k_9','k_{10}'};
    set(gcf,'Visible','off')              % turns current figure
"off"
    set(0,'DefaultFigureVisible','off');  % all subsequent figures
"off"
    e = figure;
    %interactionplot(SSValues_V,k,'varnames',{'k_3','k_4'})
    interactionplots(SSValues_V,k,'varnames',interactionstr)

    f = figure;
    interactionplots(SSValues_D,k,'varnames',interactionstr)

    g = figure;
    interactionplots(SSTime,k,'varnames',interactionstr)

    print(e,'-dpng', num2str(901));  % Prints graph to png file, i
as fn.
    print(f,'-dpng', num2str(902));  % Prints graph to png file, i
as fn.
    print(g,'-dpng', num2str(903));  % Prints graph to png file, i
as fn.

end
function PlotInteractions_ICs(SSValues_V,SSValues_D,SSTime,iC)
   interactionstr = {'L_1','R','L_2','I','Src','pSrc','G_{aby}-
GDP',...
                     'G_aGTP','G_aGDP','GEF-GTP','GAP-GDP','AB'};

    set(gcf,'Visible','off')              % turns current figure
"off"
    set(0,'DefaultFigureVisible','off');  % all subsequent figures
"off"
    e = figure;
    interactionplots(SSValues_V,iC,'varnames',interactionstr)
    f = figure;
    interactionplots(SSValues_D,iC,'varnames',interactionstr)
    g = figure;
    interactionplots(SSTime,iC,'varnames',interactionstr)

    print(e,'-dpng', num2str(1001));  % Prints graph to png file, i
as fn.
    print(f,'-dpng', num2str(1002));  % Prints graph to png file, i
as fn.
    print(g,'-dpng', num2str(1003));  % Prints graph to png file, i
as fn.

end
```

InteractionPlots

```
function  [H, ax, BigAx]  = interactionplots(y,group,varargin)
%INTERACTIONPLOT Interaction plot for grouped data
%   INTERACTIONPLOT(Y,GROUP) displays the two-factor interaction
plot for
%   the group means of matrix Y with groups defined by entries in
the cell
%   array GROUP.  Y is a numeric matrix or vector.  If Y is a
matrix, the
%   rows represent different observations and the columns represent
%   replications of each observation.  If Y is a vector, the rows
give the
%   means of each entry in the cell array GROUP.  Each cell of GROUP
must
%   contain a grouping variable that can be a categorical variable,
numeric
%   vector, character matrix, or single-column cell array of
strings.
%   GROUP can also be a matrix whose columns represent different
grouping
%   variables.  Each grouping variable must have the same number of
rows as
%   Y.  The number of grouping variables must be greater than 1.
%
%   The interaction plot is a matrix plot, with the number of rows
and
%   columns both equal to the number of grouping variables. The
grouping
%   variable names are printed on the diagonal of the plot matrix.
The
%   plot at off-diagonal position (i,j) is the interaction of the
two
%   variables whose names are given at row diagonal (i,i) and column
%   diagonal (j,j), respectively.
%
%   INTERACTIONPLOT (...,'PARAM1',val1,'PARAM2',val2,...) specifies
one or
%   more of the following parameter name/value pairs:
%
%       Parameter    Value
%       'varnames'   Grouping variables names in a character matrix
or
%                    a cell array of strings, one per grouping
variable
%                    (default names are 'X1', 'X2', ...)
%       'full'       A logic value true (default) or false. When
full is
%                    true, the matrix plot includes interaction
plots for
%                    AB and BA where A and B are any two factors in
GROUP.
%                    When full is false, only interaction plot for
AB is
%                    plotted.
%
```

```
%   [H,AX,BIGAX] = INTERACTIONPLOT(...) returns a handle H to the
figure
%   window, a matrix AX of handles to the subplot axes, and a handle
%   BIGAX to the big (invisible) axes framing the subplots.
%
%   Example:
%     Display interaction plots for data with four 3-level factors
named
%     'A', 'B','C', and 'D'.
%        y = randn(1000,1); %response
%        group = ceil(3*rand(1000,4)); %four 3-level factors
%        interactionplot(y,group,'varnames',{'A','B','C','D'})
%
%   See also MAINEFFECTSPLOT, MULTIVARICHART

% Copyright 2006-2011 The MathWorks, Inc.

if nargin <2
    error(message('stats:interactionplot:FewInput'))
end

% transpose y if it is row vector
if size(y,1) ==1
    y = y(:);
end;

% parse parameter/value pairs
args =   {'varnames','full'};
defaults = {'',true};
[varnames,full] =
internal.stats.parseArgs(args,defaults,varargin{:});

if ~iscell(varnames) && ~ischar(varnames)
    error(message('stats:interactionplot:BadVarnames'))
end
if (~(ischar(varnames) || iscellstr(varnames)))
      error(message('stats:interactionplot:BadVarnames'));
end

% determine whether we need default group variable names
needvarnames = isempty(varnames);

% convert the  numerical GROUP to cell arrays
if  isnumeric(group)
    group = num2cell(group,1);
end

group = group(:);
ng = length(group); % number of grouping factors

% You cannot have only one factor
if ng<2
    error(message('stats:interactionplot:TooFewFactors'))
end
```

```
% Convert numerical cells or char cells to string cells
for i = 1:ng
    if ischar(group{i})
        group{i} = cellstr(group{i});
    end
end
% Group variable should have the same number of items as y.
if  any(cellfun(@length,group)~=size(y,1))
    error(message('stats:interactionplot:BadGroup'))
end;

% Convert all grouping variables to integers, and remember
separately their
% original names
levelnames = cell(1,ng);
for i = 1:ng
    if isnumeric(group{i})
        if ~isvector(group{i})
           error(message('stats:multivarichart:BadGroup'))
        end
    end;
    [group{i},levelnames{i}] = grp2idx(group{i});
end

if  needvarnames
    % generate default varnames
    varnames = strcat({'X'},num2str((1:ng)','%d'));
end;

% Convert character matrix to cell array
if ischar(varnames)
    varnames = cellstr(varnames);
end;

% the length of varnames should be the same as the number of
grouping factors
if ng ~= length(varnames)
    error(message('stats:interactionplot:MismatchVarnameGroup'))
end;

% get means across replications
ybar = nanmean(y,2);

% plotting starts here
clf;
BigAx = newplot;
hold_state = ishold;
set(BigAx,'Visible','off','color','none')

% Create and plot into axes
if full % full matrix form
    % full plot is an ng by ng matrix plot.
    rows = ng;
    cols = ng;
    ax = zeros(rows,cols);
```

```
    pos = get(BigAx,'Position');
    % width and height for each individual axes
    width = pos(3)/(cols+1);
    height = pos(4)/rows;
    space = .15; % 2 percent space between axes
    %space = .02; % 2 percent space between axes
    % the position of the big axes is adjusted
    pos(1:2) = pos(1:2) - .05*[ng*width/2 height];
    ylim = nan(rows,cols,2);
    % this is the x coordinate for the legends
    if ng == 2
        legx = pos(1) + pos(3) - 1.6*width/ng;
    else
        legx = pos(1) + pos(3) - 2*width/ng;
    end
    for i=rows:-1:1,
        for j=cols:-1:1,
            axPos = [pos(1)+(j-1)*width pos(2)+(rows-i)*height ...
                width*(1-space) height*(1-space)];    % position of
each panel axes
            ax(i,j) = axes('Position',axPos, 'visible', 'on',
'Box','on');
            if  i~=j   % off- diagonal are filled with interaction
plots

plotaninteraction(ybar,group{j},group{i},varnames{j},varnames{i},...
                    levelnames{j},levelnames{i});
                ylim(i,j,:) = get(gca, 'ylim');
            else
                % make an invisible interaction plot so that I can
make a
                % legend on the diagonal.
                idx = i;          % factor to be legend
                anotheridx = mod(j,ng)+1;  % just another factor
                handles =
plotaninteraction(ybar,group{anotheridx},group{idx},...
                    varnames{anotheridx},varnames{idx},...
                    levelnames{anotheridx},levelnames{idx});
                set(handles,'visible','off')
                set(gca,'xticklabel','','yticklabel','', ...
                    'xtick',[],'ytick',[])
                % make legend texts
                levels = levelnames{idx};
                left = [varnames{idx}, '= '];
                lentext = strcat({left},levels);
                % make the legend
                %legh =
legend(lentext,'FontSize',6,'location','northeast');
                legh =
legend(lentext,'FontSize',6,'location','south');
                % place the legend to the very right
                legpos = get(legh, 'position');
                legpos(1) = legx;
                set(legh, 'position',legpos)
            end
        end
    end
```

```
    % find the best ylim
    ylimmin = min(min(ylim(:,:,1),[],2));
    ylimmax = max(max(ylim(:,:,2),[],2));

    % put the xticklabel to top in the top-row axes
    set(ax(1,:),'XAxisLocation','top');

    % put the yticklabel to top in the most right axes
    set(ax(:,cols),'YAxisLocation','right')

    % Ticks and labels on outer plots only
    set(ax(2:rows-1,:),'xticklabel','')
    set(ax(:,2:cols-1),'yticklabel','')

set(BigAx,'XTick',get(ax(rows,1),'xtick'),'YTick',get(ax(rows,1),'yt
ick'), ...
        'userdata',ax)
else % compact matrix form
    % figure out how many rows and cols are needed.
    if mod(ng,2)==0
        rows = ng/2;
        cols = ng -1;
    else
        cols = ng;
        rows = (ng -1)/2;
    end;
    ax = zeros(rows,cols);
    pos = get(BigAx,'Position');
    width = pos(3)/cols;
    height = pos(4)/rows;
    % try to work out spaces between axes
    switch ng
        case 2
            space = 0;   % no space is needed if there is only a
single plot
        case 3
            space = [0.02 0];  % no vertical space is needed if
there is only one row
        otherwise
            space = [.02 .15]; % 2 percent space between x axes and
15 percent between y axes
            pos(1:2) = pos(1:2) + space.*[width height/2];
    end;
    ylim = nan(ng*(ng-1)/2,2);
    plotind = 0;
    for i = 1:ng-1
        for j = i+1:ng
            plotind = plotind + 1;          % plot sequence number
            rowid  = ceil(plotind/cols);    % row number
            colid = mod(plotind-1, cols)+1; % col number
            axPos = [pos(1)+(colid-1)*width pos(2)+(rows-
rowid)*height ...
                [width height].*(1-space)];    % position of each
panel axes
            ax(rowid, colid) = axes('Position',axPos, 'visible',
'on', 'Box','on');
```

```
plotaninteraction(ybar,group{i},group{j},varnames{i},varnames{j},lev
elnames{i},levelnames{j});
            xlab = xlabel(varnames{i});
            set(xlab,'FontSize',10);
            ylim(plotind,:) = get(gca, 'ylim');
            levels = levelnames{j};
            left = [varnames{j}, ' = '];
            lentext = strcat({left},levels);
            if ng == 2
                legend(lentext,'FontSize',8,'location','best'); %
special treatment for single plot
            else

legend(lentext,'FontSize',8,'location','southoutside');
            end;
        end
    end;
    ylimmin = min(ylim(:,1));
    ylimmax = max(ylim(:,2));
    set(ax(:,2:cols),'yticklabel','')
end;

set(ax, 'xgrid','off', 'ygrid','off') % set axes grids off

% Set all the limits to be the same and leave
% just a 5% gap between data and axes.
inset = .05;
dy = (ylimmax - ylimmin)*inset;
set(ax,'ylim',[ylimmin-dy ylimmax+dy])

if full
    % place the variable names on the diagonal of the matrix
    for j=1:cols
        set(gcf,'CurrentAx',ax(j,j));
        xlims = get(gca,'xlim');
        ylims = get(gca,'ylim');
        h = text(mean(xlims), mean(ylims), ...
            varnames{j},
'HorizontalAlignment','center','VerticalAlignment','middle');
        set(h, 'fontsize',12)
    end
end

% Make BigAx the CurrentAxes
set(gcf,'CurrentAx',BigAx)
if ~hold_state,
    set(gcf,'NextPlot','replace')
end

% Also set Title and X/YLabel visibility to on and strings to empty
set([get(BigAx,'Title'); get(BigAx,'XLabel'); get(BigAx,'YLabel')],
...
    'String','','Visible','on')

% Return the figure handle if it is asked.
```

```
if nargout>0
    H = gcf;
end;


%-----------------------------------
function   handle =
plotaninteraction(y,factor1,factor2,varname1,varname2,levels1,levels
2)
% plot an single interaction plot between two factors
% factor1 is for x axis
% factor2 is for y axis
[interact,num] = grpstats(y,{factor1,factor2}, {'mean','numel'});  %
group means w.r.t the two factors

%number of levels in each factor
num1 = length(levels1);
num2 = length(levels2);

if  length(num) < num1*num2
    error(message('stats:interactionplot:UnequalLevels'))
end

% the means are reshaped as a matrix for the convenience of plot and
legend
matrixdata = reshape(interact,num2,num1);

% plot this matrix data
%linetype = {'-',':','-.','--'};  % all line types
linetype = {'-','-','-','-'};  % all line types

colors = [0     0     1          % all colors
          0    .5     0
          1     0     0
          0    .75   .75
         .75    0    .75
         .75   .75    0
         .25   .25   .25];
nlinetype = length(linetype);
ncolors = size(colors,1);
hold on
handle = zeros(num2,1);
for i = 1:num2
    idxline = mod(i-1,nlinetype)+1;  %  cycle through line types
    idxcolor = mod(i-1,ncolors)+1;   %  cycle through line colors
    linespec = linetype{idxline};
    handle(i) =
plot(1:num1,matrixdata(i,:),linespec,'color',colors(idxcolor,:),'Lin
eWidth',1.5);
    set(gca,'FontSize',8); % Setting the font for x-label tick
marks.
end;
hold off
axis tight
```

```
% Set the x axis limit
xlim = get(gca,'xlim');
%inset = .2;
inset = .35;
df = diff(xlim)*inset;
set(gca,'xtick',1:num1, 'xticklabel',levels1,'xlim',[xlim(1)-df,
xlim(2)+df]);
box on
```

Bibliography

1. Asthagiri, Anand R. and Douglas A. Lauffenburger. "Bioengineering Models of Cell Signaling." Annual Review of Biomedical Engineering 2 (2000): 31-53.
2. Box, G. E., Hunter,W.G., Hunter, J.S., Hunter,W.G., "Statistics for Experimenters: Design, Innovation, and Discovery", 2nd Edition, Wiley, 2005, ISBN 0-471-71813-0
3. C. T. Kelley, Solving Nonlinear Equations with Newton's Method, no 1 in Fundamentals of Algorithms, SIAM, 2003. ISBN 0-89871-546-6.
4. Cooke SF, Bliss TV (2006). "Plasticity in the human central nervous system". Brain 129 (Pt 7): 1659–73.
5. D. Rymer and T. Good. Amyloidogenic β(1-40), β (25-35) and Bovine Calcitonin Are Toxic Via A Common Pathway Involving G Protein Activation. J. Biol. Chem., 276: 2523-2530, 2001
6. Dickson DW. 1997. The Pathogenesis of Senile Plaques. Journal of Neuropathology & Experimental Neurology 56(4):321-339.
7. Erik S. Welf, Ulhas P. Naik, Babatunde A. Ogunnaike, A Spatial Model for Integrin Clustering as a Result of Feedback between Integrin Activation and Integrin Binding, Biophysical Journal, 103 1379-1389.
8. Ferri, Cleusa P., Martin Prince, Carol Brayne, Henry Brodaty, Laura Fratiglioni, Mary Ganguli, Kathleen Hall, Kazuo Hasegawa, Hugh Hendrie,

Yueqin Huang, Anthony Jorm, Colin Mathers, Paulo R. Menezes, Elizabeth Rimmer, and Marcia Scazufca. "Global prevalence of dementia: a Delphi consensus study." The Lancet 366 (December 2005): 2112-2117.

9. G. Dhawan, C.K. Combs, Inhibition of Src kinase activity attenuates amyloid associated microgliosis in a murine model of Alzheimer's disease, Journal of Neuroinflammation, 9 (2012).
10. Gilmore, Andrew P., Anthony D. Metcalfe, Lewis H. Romer, and Charles H. Streuli. "Integrin-Mediated Survival Signals Regulate the Apoptotic Function of Bax through Its Conformation and Subcellular Localization." The Journal of Cell Biology 149 (April 2000): 431-446.
11. H. Berry, V. Larreta-Garde, Oscillatory behavior of a simple kinetic model for proteolysis during cell invasion, Biophys J, 77 (1999) 655-665.
12. H.A. Pearson, C. Peers, Physiological roles for amyloid β peptides, The Journal of physiology, 575 (2006) 5-10.
13. H.J. Ahn, D. Zamolodchikov, M. Cortes-Canteli, E.H. Norris, J.F. Glickman, S. Strickland, Alzheimer's disease peptide beta-amyloid interacts with fibrinogen and induces its oligomerization, Proceedings of the National Academy of Sciences of the United States of America, 107 (2010) 21812-21817.
14. Hamby, D. M. "A Comparison of Sensitivity Analysis Techniques." Health Physics 68 (February 1995): 195-204.
15. Hardy J, Selkoe DJ. 2002. The amyloid hypothesis of Alzheimer's disease: progress and problems on the road to therapeutics. Science 297:353-356

16. Hoops, Stefan, Sven Sahle, Ralph Gauges, Christine Lee, Jürgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. "COPASI—a COmplex PAthway SImulator." Bioinformatics 22 (December 2006): 3067-3074.
17. Huveneers, Stephan and Erik H. J. Danen. "The Interaction of Src Kinase with β3 Integrin Tails: A Potential Therapeutic Target in Thrombosis and Cancer." The Scientific World JOURNAL 10 (2010): 1100-1106.
18. Hynes, Richard O. "Integrins." Cell 110 (September 2002): 673-687.
19. Ittner, Lars M., Yazi D. Ke, Fabien Delerue, Mian Bi, Amadeus Gladbach, Janet van Eersel, Heidrun Wölfing, Billy C. Chieng, MacDonald J. Christie, Ian A. Napier, Anne Eckert, Matthias Staufenbiel, Edna Hardeman, and Jürgen Götz. "Dendritic Function of Tau Mediates Amyloid-β Toxicity in Alzheimer's Disease Mouse Models." Cell 142 (August 2010): 387-397.
20. Jones, Kenneth A. and Robert W. Baughman. "Both NMDA and non-NMDA subtypes of glutamate receptors are concentrated at synapses on cerebral cortical neurons in culture." Neuron 7 (October 1991): 593-603.
21. Kalia, Lorraine V., Jeffrey R. Gingrich, and Michael W. Salter. "Src in synaptic transmission and plasticity." Oncogene 23 (October 2004): 8007-8016.
22. Kitano, Hiroaki. "Systems Biology: A Brief Overview." Http://science.sciencemag.org/content/295/5560/1662. Science, 1 Mar. 2002. Web. 12 Feb. 2016. <http://science.sciencemag.org/content/295/5560/1662.full>.

23. Klafki, H.-W., Staufenbiel, M., Kornhuber, J., Wiltfang, J., 2006. Therapeutic approaches to Alzheimer's disease. Brain, 129, 2840-2855, doi:10.1093/brain/awl280.

24. Kudryashov, Nikolay A. and Anastasia S. Zakharchenko. "Analytical properties and exact solutions of the Lotka–Volterra competition system." Applied Mathematics and Computation 254 (March 2015): 219-228.

25. Lee, Chuang-Chung, Arpan Nayak, Ananthakrishnan Sethuraman, Georges Belfort, and Gregory J. McRae. "A Three-Stage Kinetic Model of Amyloid Fibrillation." Biophysical Journal 92 (May 2007): 3448-3458.

26. Li, Shaomin, Ganesh M. Shankar, and Dennis J. Selkoe. "How do Soluble Oligomers of Amyloid beta-protein Impair Hippocampal Synaptic Plasticity?." *Frontiers in cellular neuroscience* 4 (2010).

27. M.W. Hirsch, S. Smale, and R.L. Devaney (2004). Differential Equations, Dynamical Systems and an Introduction to Chaos. Academic Press/Elsevier, San Diego.

28. Nussbaum, J. M.; Seward, M. E.; Bloom, G. S. (2013). "Alzheimer disease: A tale of two prions". Prion 7 (1): 14–9.

29. Ota, Takahide, Masayo Maeda, Mayumi Okamoto, and Masaaki Tatsuka. "Positive Regulation of Rho GTPase Activity by RhoGDIs as a Result of Their Direct Interaction with GAPs." BMC Systems Biology BMC Syst Biol 9.1 (2015): 3. Web.

30. Radhakrishnan, K., J. S. Edwards, D. S. Lidke, T. M. Jovin, B. S. Wilson, and J. M. Oliver. "Sensitivity analysis predicts that the ERK-pMEK interaction regulates ERK." IET Systems Biology 3 (September 2009): 329-341.
31. Sabo, S., M. P. Lambert, K. Kessey, W. Wade, G. Krafft, and W. L. Klein. "Interaction of beta-amyloid peptides with integrins in a human nerve cell line." Neuroscience Letters 184 (January 1995): 25-28.
32. Shan, Ruiqin and Tianfeng Lu. "Ignition and extinction in perfectly stirred reactors with detailed chemistry." Combustion and Flame 159 (June 2012): 2069-2076.
33. Shankar, Ganesh M., Brenda L. Bloodgood, Matthew Townsend, Dominic M. Walsh, Dennis J. Selkoe, and Bernardo L. Sabatini. "Natural Oligomers of the Alzheimer Amyloid-β Protein Induce Reversible Synapse Loss by Modulating an NMDA-Type Glutamate Receptor-Dependent Signaling Pathway." *The Journal of Neuroscience* 27 (March 2007): 2866-2875.
34. Shen, Bo, M. Keegan Delaney, and Xiaoping Du. "Inside-out, outside-in, and inside-outside-in: G protein signaling in integrin-mediated cell adhesion, spreading, and retraction.." Current opinion in cell biology 24 (October 2012): 600-606.
35. Steuer, R., and B. H. Junker. Computational Models of Metabolism: Stability and Regulation in Metabolic Networks. Vol. 142. Hoboken, NJ: John Wiley, 2009. PDF.

36. Steuer, Ralf, Thilo Gross, Joachim Selbig, and Bernd Blasius. "Structural kinetic modeling of metabolic networks." Proceedings of the National Academy of Sciences 103 (August 2006): 11868-11873.
37. Steuer, Ralf. "Computational approaches to the topology, stability and dynamics of metabolic networks." Phytochemistry 68 (August 2007): 2139-2151.
38. T. Good and R. Murphy, Aggregation State-Dependent Binding of b - Amyloid Peptide to Protein and Lipid Components of Rat Cortical Homogenates. Biochemical And Biophysical Research Communications, 207: 209-215, 1995.
39. Thathiah, Amantha and Bart De Strooper. "The role of G protein-coupled receptors in the pathology of Alzheimer's disease." Nat Rev Neurosci 12 (February 2011): 73-87.
40. Venkatasubramaniam, A., Drude, A., and Good, T. Role of N-terminal residues in Aβ interactions with integrin receptor and cell surface. BBA Biomembranes, 1838: 2568-77, 2014.
41. Venkatasubramaniam, Arundhathi. "ENGINEERING TOOLS TO ANALYZE BETA-AMYLOID'S EFFECT ON CELLULAR MECHANISMS LINKED TO THE NEURODEGENERATION OBSERVED IN ALZHEIMER'S DISEASE." Thesis. University of Maryland, Baltimore County, 2014. Print.
42. Wang, H.-W., Pasternak, J. F., Kuo, H., Ristic, H., Lambert, M. P., Chromy, B., Viola, K. L., Klein, W. L., Stine, W. B., Krafft, G. A., Trommer, B. L.,

2002. Soluble oligomers of beta-amyloid (1-42) inhibit long-term potentiation but not long-term depression in rat dentate gyrus. Brain Research, 924, 133.

43. Zhang, Chi, Qiu, Grant A. Krafft, and William L. Klein. "Protein kinase C and F-actin are essential for stimulation of neuronal FAK tyrosine phosphorylation by G-proteins and amyloid beta protein." FEBS Letters 386 (May 1996): 185-188.