Fall 9-14-2017

Exploring biological heterogeneity and its consequences at tissue and cellular scales through mathematical and computational modeling

Romica Kerketta
University of New Mexico

ROMICA KERKETTA

Candidate

BIOMEDICAL SCIENCES- PATHOLOGY

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

JEREMY S. EDWARDS, Ph.D., Chairperson

BRIDGET S. WILSON, Ph.D.

DIANE S. LIDKE, Ph.D.

VITTORIO CRISTINI, Ph.D.

ZHIHUI WANG, Ph.D.

EXPLORING BIOLOGICAL HETEROGENEITY AND ITS CONSEQUENCES AT TISSUE AND CELLULAR SCALES THROUGH MATHEMATICAL AND COMPUTATIONAL MODELING

by

ROMICA KERKETTA

B.S., Cell and Molecular Biology, San Francisco State Univ., 2010

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Biomedical Sciences

The University of New Mexico
Albuquerque, New Mexico

December 2017

DEDICATION

I dedicate this dissertation to my wonderful parents, my supportive brother and my awesome husband without whose love and support this journey would have been immeasurably harder.

To my mother, for always encouraging me to work hard and give my best every step of the way. Thank you mom for being my inspiration and motivating me to reach for the stars. To my father, for showering his constant love and care on me and for the many sacrifices that he made which helped me reach here today. To my brother Ranjan, for looking after me and supporting me always, especially when mom and dad were not around in San Francisco. To my loving husband Vishvas, who has been my rock and one of my life's greatest blessings, for holding my hand when the going got tough, for laughing with me when the tough got going, for guiding and believing in me. Thank you for never letting me fall and always being there for me.

ACKNOWLEDGEMENTS

I would like to acknowledge all my committee members for their constant support and encouragement. To my mentor, Dr. Jeremy Edwards, I will forever be grateful for providing me an opportunity to work with you. Your support and encouragement have been crucial in the completion of my PhD. To Dr. Bridget Wilson, I am truly indebted for your guidance and critical feedback throughout my PhD journey. You have been instrumental in my success and I am filled with immense gratitude for you. To Dr. Diane Lidke, thank you for supporting me, giving me your feedback and helping me figure out many of the answers to my research questions. To Dr. Vittorio Cristini, thank you for giving me an excellent start to my journey of becoming a computational biologist. To Dr. Zhihui Wang, thank you for your support, feedback and guidance.

I will truly be grateful to Dr. Adam Halasz for going above and beyond and providing me altruistic and invaluable guidance throughout my PhD. Thank you for being an amazing mentor to me. I could not have come this far without your vision, feedback and support.

To Dr. Michael Wester for helping me when I first transitioned into STMC, answering my programming questions and, most importantly, for his wonderful friendship. To Dr. Mara Steinkamp for her critical feedback and support during preparation of manuscripts. To Dr. M. Frank Erasmus and Rachel Grattan for providing experimental data for modeling. To Dr. Meghan Marie Pryor for providing computational modeling support. To Edwards and Cristini lab members for their useful feedback and encouragement. To Ryan Tanner for providing administrative support at the STMC. To STMC and CNTC for my funding during my graduate studies.

EXPLORING BIOLOGICAL HETEROGENEITY AND ITS CONSEQUENCES AT TISSUE AND CELLULAR SCALES THROUGH MATHEMATICAL AND COMPUTATIONAL MODELING

by

ROMICA KERKETTA

B.S., Cell and Molecular Biology, San Francisco State University, 2010
Ph.D., Biomedical Sciences, University of New Mexico, USA, 2017

ABSTRACT

This dissertation explores the effects of heterogeneity across different biological scales in cancer as well as normal cells. At the tissue scale, we investigated the variability present in the tumor microenvironment and its effect on patient chemotherapeutic outcomes using a mathematical model of drug transport. We found that parameters such as tumor blood perfusion and radius of blood vessel had an impact on the tumor cytotoxicity. This indicated that the physical microenvironment of the tumor is an important regulator of the tumor response to chemotherapy. At the cellular scale, we investigated the heterogeneity present on the membrane landscape of ErbB2 and ErbB3, two receptors that are upregulated in cancer, using a spatial stochastic model of receptor dimerization and phosphorylation. We found that membrane domains played an important role in regulating signaling emanating from this receptor dimer. In our next study, we developed a 3-D spatial stochastic model of pre-BCR, a receptor which is crucial in the development of B lymphocytes and also upregulated in a subset of patients with B-Cell Precursor Acute Lymphoblastic Leukemia, to investigate the effects of ligand independent (tonic signaling) originating from this receptor. We populated our model with single particle tracking data from two different leukemic cell lines which had different dimer off rates and diffusion coefficients, along with experimental measurements. Other important signaling molecules such as Lyn and Syk, which are active in this pathway, were also included in the model. We found that the variability in characteristics between the two cell lines led to differences in downstream signaling events from the receptor. The cell line with the lower dimer off rate formed higher order oligomers and had more overall molecule phosphorylation compared to the other. Thus, this spatial stochastic model was able to shed light on threshold signaling events which take place during tonic signaling.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

Heterogeneity in form and function has been found to exist across multiple different scales in biological systems (Altschuler and Wu, 2010; Allison and Sledge, 2014; Chang and Marshall, 2017). At the cellular scale, clonally identical cell populations have been found to exhibit differential gene expression of the same protein, thus giving rise to phenotypic diversity (Elowitz et al., 2002; Ozbudak et al., 2002). Stochastic generation and degradation of proteins and compartmentalization of molecules during cell division also contribute to non-genetic sources of heterogeneity in cells (Huh and Paulsson, 2011). At the organelle scale, rates of biochemical reactions occurring inside structures such as the mitochondria and endoplasmic reticulum can be affected by the size and shape of these structures (Marshall, 2012). At the tissue scale, the geometrical shapes of mammary epithelial tubules have been found to affect the positioning of ductal branches during pubertal mammary morphogenesis (Nelson et al., 2006). A computational model of prostate cancer exploring tumor growth has found that malignancy of tumors is affected by the geometry of the tumors, as well as the anatomy of the organ in which the tumor resides (Lorenzo et al., 2016). Thus, heterogeneity has been found to exist across all scales in normal as well cancerous tissues (Editorial, 2010).

1.2 MOTIVATION

Since the existence of heterogeneity is well established, this poses some interesting questions such as to whether they are harmless or they render cells or groups of cells with evolutionary or survival advantages (Editorial, 2010; Ackermann, 2015). Understanding the consequences of such heterogeneity is particularly important in diseases like cancer as tumor cells can exploit heterogeneity present in the tumor micro-environment for growth and escape from patient therapies (Junttila and de Sauvage, 2013; Allison and Sledge, 2014). Even patients with the same type of cancer can inherit or develop different tumor heterogeneities leading to varied response to chemotherapeutic drugs (Harry et al, 2010.; Tonkin et al., 1985; Wei et al., 2013). Thus, for optimal treatment outcomes, it is important to not only understand the consequences of such noise but also to develop methods to identify the heterogeneities as they can differ from patient to patient (Meacham and Morrison, 2013; Pascal et al., 2013a; Wang et al., 2016). Conversely, heterogeneity can also be a beneficial trait in normal cells, for example, non-homogenous distribution of molecules or receptors in different domains on the plasma membrane can help to regulate signal transduction pathways in the cell (Lagerholm et al., 2005).

In this dissertation, we explore the heterogeneity present on the tissue scale (macroscopic) as well as at the cellular scale (microscopic) in the context of cancer. In Chapter 1, we explore the effects of heterogeneity present in the tumor micro-environment of patients and its effect on the outcome of chemotherapy. The heterogeneity that exists is dependent upon the amount of blood vessel perfusion of the tumor as well as the geometry of the blood vessels. Using a mathematical model of drug transport, we predicted the fraction of dead tumor in patients who had been administered chemotherapy. We then retrospectively compared it to the actual fraction of dead tumor measured in these patients to validate the model. Patient histological samples that displayed higher levels of blood vessel perfusion also exhibited higher fractions of dead tumor indicating that the patient vasculature, which is inherently varied between patients, affects treatment outcomes. Thus, this model highlighted the effects of a heterogeneous tumor microenvironment on tumor survival and consequently patient therapeutic outcomes.

In Chapter 2, we investigate the effects of a heterogeneous membrane landscape on the signal transduction pathways of ErbB2 and ErbB3, two receptors belonging to the epidermal growth factor receptor (EGFR) family. These two receptors are often found upregulated in cancer and together form a potent oncogenic unit by activating key survival and proliferative cellular pathways. We found that in normal cells, spatial segregation of these two receptors into different *in silico* membrane domains downregulates the signals arising from dimer events between them. Additionally, we also found that strength of the confinement of domains affected receptor signaling based on the amount of overlap between the two receptor domains. Thus, in a non-cancerous setting, membrane domains add an additional layer of regulation of the potent ErbB2/ErbB3 signaling pathway.

In Chapter 3, our focus was on generating a mathematical model of tonic signaling (ligand independent signaling) arising from the pre-BCR. The pre-BCR is expressed early in the developmental pathway of B lymphocytes, where it is crucial for the survival and differentiation of progenitor B lymphocytes. This receptor is also characteristic of a subset of patients with B-Cell Precursor Acute Lymphoblastic Leukemia (BCP-ALL), where the tumor exploits the tonic signaling pathway for its survival and proliferation. Single particle tracking (SPT) methods have revealed that these receptors have transient, but frequent dimerization events with each other. We investigated tonic signaling emanating from this receptor using two different BCP-ALL cell lines and found characteristic differences between them in diffusion coefficients and dimer off rates. We created a spatial stochastic model of pre-BCR aggregation to explore the membrane landscape of pre-B cells during tonic signaling in more detail. We found that the individual differences along with the presence of membrane domains impacted aggregate size, receptor phosphorylation and downstream activation of signaling molecules in both the cell lines.

The overall goal of this work was to explore heterogeneity present at different biological scales and its impact on behavior of cells or tissues in cancerous or normal settings. Below is a more detailed description of the heterogeneity present in the tissue scale as well as the cellular scales in particular biological systems.

1.3 HETEROGENEITY AT THE TISSUE SCALE IN TUMOR MICROENVIRONMENT

Genetic variability has been established as a key feature among tumor cells and has been identified as the primary driver of oncogenic mutations (Hanahan and Weinberg, 2000; Meacham and Morrison, 2013). However, research in the past two decades has alluded to us other accessory participants that promote this cancerous phenotype, which include cells in the surrounding stroma as well as the vasculature and the lymphatic system (Hashizume et al., 2000; Nagy et al., 2009; Goel et al., 2011; Hanahan and Weinberg, 2011). Lymphocytes such as T and B cells and tumor associated macrophages (TAMs) have been found to be associated with tumor cells and a certain subset of these cells such as the T regulatory (Tregs) and B regulatory cell (Bregs) have displayed tumor promoting capabilities by downregulating anti-tumor responses (Balkwill et al., 2012). Quiescent vascular endothelial or lymphatic endothelial cells can be activated to produce new blood or lymphatic vessels through the binding of growth factors secreted by the tumor cells as wells as through a hypoxic tumor microenvironment (Balkwill et al., 2012). This ensures the tumor a supply of nutrients while it proliferates and helps it to metastasize to distant locations in the body. Thus, the tumor microenvironment forms a dynamic heterogeneous spatial landscape, whose interactions with the tumor cells need to be investigated thoroughly as the microenvironment can significantly impact tumor growth and alter patient responses to therapies (Junttila and de Sauvage, 2013; Yuan, 2016).

In vitro and *in vivo* studies have shown that the microenvironment of the tumor plays a key role in drug penetration into the tumor and might be a potential reason for chemotherapeutic resistance or failure in patients (Kuh et al., 1999; Tunggal et al., 1999; Tannock et al., 2002; Kyle et al., 2004; Primeau et al., 2005; Grantab et al., 2006; Kyle et al., 2007; Sinek et al., 2009; Grantab and Tannock, 2012; Rejniak et al., 2013). Systemic delivery of drugs into the tumor sites involves the transport of drugs through the tumor vasculature, extravasation across the blood vessel and transport across the interstitium within the tumor (Jain, 1989; 2005). There are three main physiological barriers that hinder the transportation of drugs into the tumor and these are variations in the amount of blood supplied to the tumor, an increase in the interstitial fluid pressure (IFP) and large transport distances travelled by tumor drugs to reach the tumor site. These barriers play an important role in drug delivery as they can limit the amount of drug transported into the tumor, thus, reducing the effectiveness of the chemotherapeutic treatment, leading to residual tumor cells and cancer regrowth (Au et al., 2001; Gottesman, 2002; Jang et al., 2003; Minchinton and Tannock, 2006; Sanga et al., 2006; Junttila and de Sauvage, 2013). A heterogeneous vasculature in the tumor microenvironment can give rise to necrotic, semi-necrotic and well vascularized regions with irregular vessel diameters and leaky vessels leading to lower blood flow to the tumor site (Hobbs et al., 1998; Jang et al., 2003; Minchinton and Tannock, 2006; Tredan et al., 2007; Stylianopoulos and Jain, 2013). High IFP is often associated with solid tumors due to lack of lymphatics and leakiness of the tumor blood vessels leading to limited extravasation of drugs from the blood vessel and worse patient prognosis (Jain, 1987; Baxter and Jain, 1989; Curti et al., 1993; Milosevic et al., 2001; Jang et al., 2003; Heldin et al., 2004; Minchinton and Tannock, 2006; Sven and Josipa, 2007; Li et al., 2011). Penetration of drugs into the solid tumor has been shown to be obstructed by high tumor cell density where tumors with a lower tumor cell fraction and more interstitial space had a more rapid diffusion of drugs into the tumor (Au et al., 2001).

Systemic chemotherapy is a commonly used treatment strategy for various types of cancers (Carlson et al., 2009; Edwards et al., 2012; Kim et al., 2013; Wei et al., 2013). However, due to the existence of heterogeneity present in the microenvironment, response to chemotherapy has been variable even in the same type of cancer. For instance, response rates ranging from 8% to 85% have been observed for colorectal cancer treated with 5-fluorouracil and response rates ranging from 8% to 60 % have been observed for head and neck cancer treated with methotrexate (Tonkin et al., 1985). Thus, predicting chemotherapeutic outcomes is important to optimize treatment of cancer patients as prolonged continuation with ineffective therapies can lead to an increased burden of toxicity and expense for the patients along with a delay in surrogate beneficial treatments (Dose Schwarz et al., 2005; Harry et al., 2010). Current methods for assessing tumor response to chemotherapy include unidimensional response evaluation criteria to solid tumors (RECIST) through the use of tumor imaging techniques such as magnetic resonance imaging (MRI), computed tomography (CT), ultrasound, chest X-ray etc. (Therasse et al., 2000). These methods have been modified from the previous bidimensional measurements recommended by the World Health organization (WHO) (Organization, 1979). However, many of the techniques used for tumor assessment are proven to be inadequate, while others can only be utilized after sufficient time has passed at which changes in tumor can be evaluated (Rubbia-Brandt et al., 2007; Glazer et al., 2010; Thoeny and Ross, 2010; Egger et al., 2013). Hence, there is a lack of robust quantitative measurements of tumor response that can predict chemotherapeutic outcomes even before the commencement of therapeutic regimen for patients.

In chapter 2 of this dissertation, we employ a “mathematical pathology” approach, a translational modeling approach emphasizing the development of mechanistic models that are able to predict chemotherapeutic outcomes dependent on patient-specific measurable parameters.

1.4 HETEROGENEITY AT THE MEMBRANE LEVEL

Cell signaling is initiated through ligand-receptor or receptor-receptor binding on a cell’s surface. This surface known as the plasma membrane serves as a platform for initiating signaling events that have a variety of downstream effects on cell fate and behavior (Groves and Kuriyan, 2010). Critical decisions to grow, survive, metastasize or undergo apoptosis are relayed through the plasma membrane into the cells (Radhakrishnan et al., 2012). Given the role that these surfaces play in cell signaling, it is important to investigate how the membrane themselves are regulated. While the composition of these membranes have been well defined, our understanding of their functioning and their effect on cell signaling remains rudimentary (Grecco et al.).

The membrane platforms were initially thought to be a homogenous signaling environment with receptors and other macromolecules randomly distributed on the cell surface. Singer and Nicolson in their landmark paper, described the membrane as a fluid mosaic of globular proteins embedded homogenously in a phospholipid by layer (Singer and Nicolson, 1972). However, it was soon observed that the plasma membrane was a heterogeneous landscape containing "domains" or "patches", ranging from 0.1 μm to 1.0 μm, which could transiently trap specific proteins and lipids (Kaizuka et al., 2007; Chung et al., 2010; Treanor et al., 2010a; Wilson et al., 2011; Radhakrishnan et al., 2012; Goñi, 2014). Domains enriched in increased levels of cholesterol and glycospingolipids are known as lipid rafts and they are estimated to have a diameter ranging from less than 0.1μm to 0.2 μm (Pike, 2003; Lidke and Wilson, 2009). Domains formed by the underlying actin cytoskeleton are known as corrals and they can range from 0.1 μm to 0.3 μm (Kusumi et al., 1993; Kusumi and Sako, 1996; Kusumi et al., 2005; Hoppe and Low-Nam, 2014). These domains can trap a variety of receptors such as the G-protein-coupled receptor (GPCRs), epidermal growth factor receptors (EGFR), platelet derived growth factor (PDGF) receptors and endothelin receptors among others (Smart et al., 1999; Pike, 2003).

Determining the role of these domains on regulating signal transduction pathways has been a major subject of interest in the field of membrane biology (Pike, 2003; Marguet et al., 2006; Day and Kenworthy, 2009; Owen et al., 2009). It has been postulated that membrane domains could positively or negatively inhibit reaction networks by compartmentalizing molecules belonging to specific pathways in different domains (Pike, 2003). Experimental evidence for confinement of receptors and their signaling molecules in domains or clusters have been found in a variety of important receptor systems such as the T cell receptor (TCR) (Bunnell et al., 2002; Douglass and Vale, 2005; Gaus et al., 2005; Lillemeier et al., 2006; Kaizuka et al., 2007; Dinic et al., 2015), the high affinity Immunoglobulin E receptor (Fc∊RI) (Field et al., 1997; Andrews et al., 2009a), the B cell receptor (BCR) (Tolar et al., 2009; Treanor et al., 2010b) and the Epidermal growth factor receptor (EGFR) family (Nagy et al., 2002; Yang et al., 2007; Chung et al., 2010).

In chapter 3 of this dissertation, we specifically investigated the consequences of a heterogeneous membrane landscape consisting of specific domains on two receptors belonging to the EGFR family- the ErbB2/HER2/NEU and ErbB3/HER3 receptors (Yarden, 2001). Other members of this family include the EGFR/ErbB1/HER1 and ErbB4/HER4 (Roskoski, 2014). The EGFR family are receptor tyrosine kinases (RTK) that contain an extracellular ligand binding domain, a transmembrane domain and an intracellular kinase domain (Ullrich and Schlessinger, 1990). There is no known ligand for ErbB2 and ErbB3 binds two ligands- Neuregulin-1 (Nrg-1) and Neuregulin-2 (Nrg-2) (Burden and Yarden, 1997; Roskoski, 2014). Seven different ligands including the epidermal growth factor (EGF) and transforming growth factor- α (TGF-α) bind to

ErbB1, while the ErbB4 also binds seven ligands including Nrg-1, Nrg-2, Neuregulin-3 (Nrg-3), and Neuregulin-4 (Nrg-4) (Roskoski, 2014). The EGFR family regulates cell growth, differentiation, apoptosis, adhesion and migration (Yarden and Sliwkowski, 2001). Specifically, inactivation or deficiency of EGFR/ErbB1 can lead to impaired epithelial growth and differentiation, affecting the development of skin, lungs, gastrointestinal tracts, kidney, liver as well as impaired neural development during embryonic or post-natal growth (Miettinen et al., 1995; Threadgill et al., 1995; Sibilia et al., 1998). Likewise, ErbB2, ErbB3 and ErbB4 have been implicated in development and differentiation of neural and cardiac tissue (Gassmann et al., 1995; Lee et al., 1995; Meyer and Birchmeier, 1995; Burden and Yarden, 1997; Liu et al., 1998). Thus, the EGFR family plays a crucial role in normal development of tissues and organs in an organism.

The EGFR family activates a variety of downstream signaling pathways that are involved in cell proliferation and survival such as the phosphatidylinositol 3-kinase (PI3)/Akt (PKB) pathway, the Ras/Raf/MEK/ERK1/2 pathway, and the phospholipase C (PLCγ) pathway (Yarden and Sliwkowski, 2001; Roskoski, 2014). Upon ligand binding, the EGFR family can form homodimers and heterodimers with each other, followed by transphosphorylation of tyrosine residues by the kinase domains of the partner receptors (Hubbard and Till, 2000). These phosphotyrosines then serve as docking sites for a wide variety of downstream signaling molecules such as the growth factor receptor bound protein 2 (Grb2), p85 and the Shc adaptor protein among others (Wilson et al., 2009; Roskoski, 2014). Amongst the four ErbB receptors, ErbB3 has a weak catalytic activity and it requires hetero-dimerization with another ErbB receptor, mainly ErbB2, for its phosphorylation and upregulation of its kinase activity (Guy et al., 1994; Zhang et al., 2009; Shi et al., 2010; Steinkamp et al., 2014).

As crucial as the ErbB receptors' roles remain in normal development, they have also been implicated in many cancers (Yarden and Sliwkowski, 2001). Overexpression of ErbB2 has been found in cancers of the breast, gastrointestinal tract, lung, ovaries, cervix and the salivary gland and has been associated with worse patient prognosis (Slamon et al., 1987; Ross and Fletcher, 1998; Yarden and Sliwkowski, 2001; Baselga and Swain, 2009). Aberrant signaling from the overexpressed ErbB2 receptors result in tumor survival and proliferation (Baselga and Swain, 2009). ErbB3 has been found to be co-expressed with ErbB2 in breast cancers, as well as melanoma, and the ErbB2/ErbB3 heterodimer has been implicated in transformations of normal cells in to tumor as well as increased cell spreading and motility (Alimandi et al., 1995; Wallasch et al., 1995; Chausovsky et al., 2000; Vaught et al., 2012; Zhang et al., 2013). ErbB3 expression in tumors have also been found to promote resistance to tyrosine kinase inhibitor therapies in patients (Sergina et al., 2007; Huang et al., 2013; Sato et al., 2013; Lee et al., 2014). Thus, the ErbB2/ErbB3 heterodimer pair is considered to be one of the most potent ErbB dimer pairs involved in carcinogenesis (Pinkas-Kramarski et al., 1996; Tzahar et al., 1996; Baselga and Swain, 2009).

Since the ErbB2/ErbB3 heterodimer plays an important role in cancer progression, we wanted to further study the dynamics of this signaling unit in the context of a heterogeneous membrane landscape. Images obtained from immunoelectron microscopy have showed that ErbB receptors localize distinctly to separate regions of the plasma membranes in breast cancer cells (Yang et al., 2007). Upon stimulation with a ligand, receptors such as ErbB2 and ErbB3 were found to be co-clustered, indicating an increase in proximity for receptor binding events to occur (Yang et al., 2007). One possible explanation for the segregation of ErbB receptors in different compartments before stimulation would be to regulate any spurious signaling that might occur due to the proximity of signaling dimers with each other (Yang et al., 2007). Yang et al. also postulated that in normal cells, separation of ErbB receptors in distinct areas or domains might serve to limit any unnecessary signaling originating from the ErbB heterodimers (Yang et al., 2007). In chapter 3, we have attempted to answer how confinement of receptors in different domains and the strength of the confinement in those domains affects ErbB signaling in an *in silico* membrane landscape.

1.5 HETEROGENEITY AT THE RECEPTOR LEVEL

The B lymphocytes belong to the adaptive immune system and perform important functions such as antibody and cytokine production and co-stimulation of T cells (LeBien and Tedder, 2008). Their development initiates in the bone marrow where progenitor B cells have to progress through various checkpoints to ensure their survival and differentiation into mature B cells (Rajewsky, 1996). The first checkpoint encountered by the progenitor B cells is at the pre-B cell stage where surface expression of a precursor-B cell receptor (pre-BCR) is required for transition into the next developmental stage (von Boehmer and Melchers, 2010). The structure of the pre-BCR consists of the two immunoglobulin heavy chains (IgH) that pair with two surrogate light chains (SLC)- λ5 and VpreB, along with the heterodimeric signaling unit Igα (CD79a) and Igβ (CD79b) (Benschop and Cambier, 1999). The immunoglobulin heavy chain gene locus undergoes somatic recombination for the rearrangement of the variable (V), diversity (D) and Junction (J) gene segments to produce a wide repertoire of antigen binding sites on the IgH (Tonegawa, 1983; Alt et al., 1984). Signaling from the transiently expressed pre-BCR is necessary for the transition of early progenitor B cells from the pre-BI to the pre-BII developmental stage, for the positive selection of early B cells that have successfully assembled the IgH and are capable of pairing with the light chains, Igα and Igβ subunits for signaling, for the negative selection of cells that react to self-antigen and could give rise to potential autoimmune responses, for allelic exclusion to occur so that only one of the alleles of the IgH gene transcribes and translates the IgH protein in a given cell and for proliferation and downregulation of the SLC in a negative feedback loop (Herzog et al., 2009; Mårtensson et al., 2010).

A recent study from Wilson’s group provided evidence for ligand independent aggregation and signaling of pre-BCRs that were mediated through their SLCs(Erasmus et al., 2016) . This signaling, also known as ‘tonic’ signaling, generates responses that lead to early B cell survival and proliferation (Monroe, 2006). Pre-BCR aggregation leads to recruitment of src family kinases (SFK) such as Fyn, Lyn and Blk which phosphorylate tyrosine residues on the immunoreceptor tyrosine-based activation motifs (ITAM) present on the Igα and Igβ. Phosphorylation of the ITAMs create further docking sites for the SFKs and Syk family kinase along with other downstream adaptor proteins such as Grb2 and BLNK. This induces activation of the PLCγ2 pathway which regulates calcium release from the endoplasmic reticulum (ER) and the entry of extracellular calcium into the cell (Kurosaki et al., 2000). Mobilization of calcium in these cells is essential for upregulation of enzymes such as calcineurin that are dependent on calcium (Geier and Schlissel, 2006). Calcineurin activates the family of transcription factors called nuclear factor of activated T cells (NFAT) proteins which are responsible for upregulation of genes involved in production of signaling proteins, cytokines, cell surface receptors and other molecules (Rao et al., 1997; Geier and Schlissel, 2006). Additionally, tonic signaling also activates the ERK/MAPK pathway for pre-B cell survival and proliferation (Fleming and Paige, 2001; Geier and Schlissel, 2006).

There is a general failure in precursor B cell development in pre-B cell acute lymphocytic leukemia (B-ALL). This can be associated with downregulation of the expression of pre-BCR on the cell surface (Rickert, 2013; Müschen, 2015). It has been suggested that up to 85% of B-ALL cases lack pre-BCR expression on the cell surface and some studies have indicated a tumor suppressor role for this receptor as reconstitution of the receptor or its signaling unit (Igα) led to apoptosis (Trageser et al., 2009; Chen et al., 2015; Müschen, 2015). However, there is a subset of ALL cases (~13.5%) where pre-BCR surface expression is maintained and the B cell tumor exploits the tonic signaling pathway to induce high expressions levels of B-cell lymphoma 6 protein (BCL6), a protein which is critical for survival and proliferation of B cells (Duy et al., 2010). In these subsets, inhibitors against Syk, Src and Btk tyrosine kinases, all important players in the tonic signaling pathway, led to apoptosis of pre-BCR+ ALL cells (Bicocca et al., 2012; Geng et al., 2015). Therefore, understanding the regulation of this pathway might lead to more molecular targets for therapeutic intervention.

In chapter 3 of this dissertation, we explore how tonic signaling emanating from two different BCP-ALL cell lines (697 and Nalm6) is affected by a heterogeneous membrane landscape as well as different propensities for pre-BCR aggregation on the pre-B surface.

1.6 SPATIAL STOCHASTIC MODELING OF BIOLOGICAL SYSTEMS

As discussed in the above sections, existence of heterogeneity across multiple biological scales is more the norm than the exception. Powerful imaging techniques such as single particle tracking (SPT) have allowed us to observe the organization of molecules on the plasma membrane and enabled measurements of diffusional dynamics and protein interactions across temporal and spatial scales (Owen et al. 2009b; Saxton and Jacobson 1997; Lidke and Wilson 2009). Parameters generated from such imaging techniques can be combined with computational modeling to gain a deeper understanding of the processes that regulate signal transduction pathways. They can also enable the production of experimentally testable hypothesis of cell fate and behavior. Modeling of molecular events that are stochastic in nature and where the molecules occupy different spatial niches can be simulated using a variety of spatial stochastic methods (Andrews, Dinh, and Arkin 2009). In this section, we will review some of the modeling techniques that are employed for spatial stochastic modeling of molecular reaction and diffusion events.

1.6.1 Spatial Gillespie method

Daniel Gillespie, in his landmark papers, presented the stochastic simulation algorithm (SSA) for simulating reaction events in spatially homogenous chemical systems (Gillespie, 1976; 1977). In the SSA, after the initialization of the chemical system, random numbers are generated to ascertain the type of reaction to occur as well as the time of the occurrence. The system is then updated with the reaction event and the time step. This process is repeated until the simulation time ends or the reactant molecules have been depleted such that no new reactions can occur. Lumsden and Stundzia extended the SSA by accounting for diffusion of molecules in a spatially inhomogeneous environment (Stundzia and Lumsden, 1996). The spatial inhomogeneity was achieved through segregation of the total volume into sub volumes or a coarse lattice, inside which the molecules could react and also diffuse across the lattice with a certain probability (Stundzia and Lumsden, 1996). Several simulation programs avail the spatial Gillespie method and include programs such as MesoRD, SmartCell and GMP (Ander et al., 2004; Hattne et al., 2005; Rodriguez et al., 2006; Andrews et al., 2010).

1.6.2 The microscopic lattice method

This method involves incorporation of reactant molecules into a much smaller lattice than the coarse lattice mentioned above, such that the lattice can have single reactants or be completely empty (Andrews et al., 2010). Simulation programs making use of this method include the GridCell and Spatiocyte (Boulianne et al., 2008; Andrews et al., 2010; Arjunan and Tomita, 2010). The reactants undergo random motion and can diffuse from their volume spaces into neighboring spaces for reaction events.

1.6.3 Particle based methods

In particle based methods, molecular species are represented as point like particles that occupy distinct spatial positions (Andrews et al., 2010). At each simulation time step, Δt, molecules are randomly diffused using "jumps" or displacements that are generated from a Gaussian probability density function (Andrews et al., 2009b). Diffusion of molecules is followed by scanning of the landscape for binding events with other molecules. If a reaction is possible, a binding event is recorded, otherwise the particles are updated in the next time step. A number of publicly available software packages that use particle based method for simulation of reaction events include packages such as Smoldyn, MCell and Green's function reaction dynamics (GFRD) (Stiles and Bartol, 2001; Andrews and Bray, 2004; van Zon and Ten Wolde, 2005; Erban, 2014). However, GFRD is different from Smoldyn and MCell in that it uses a variable time step for its reaction events (van Zon and Ten Wolde, 2005; Andrews et al., 2010; Erban, 2014). In chapters 3 and 4 of this dissertation, we implemented a spatial stochastic algorithm based on the modeling approach used in Smoldyn (extension of the Smoluchowski model), where a fixed time step is used for each particle update (Andrews and Bray, 2004; Erban, 2014). This approach has also been used previously by our group to study the dynamics of EGFR and ErbB2/ErbB3 receptors (Pryor et al., 2013; Pryor et al., 2015). Below is a description of the modeling technique used in this dissertation.

1.6.3.1 Defining the simulation space and reaction network

In our model, the 2D (chapter 3) and 3D (chapter 4) simulation spaces were generated using the software Matlab and populated with molecules with a specific density based on their cellular density and the simulation area/volume. The reactions between molecules consisted of first order reactions such as molecule dissociation, phosphorylation, dephosphorylation and domain escape reactions, as well as second order reactions such as molecular associations. The rules representing reactions events between molecules were programmed in Fortran.

1.6.3.2 Molecule diffusion

We used Brownian dynamics (BD) to simulate particle diffusion in x, y and z plane. In BD, molecules are treated as point like particles which undergo random motion and can undergo reaction events upon collision (Andrews et al., 2010). For diffusion, particle "jumps" or displacement are generated, when random numbers chosen from a normal distribution are multiplied by the root mean square (RMS) of the molecule (Andrews and Bray, 2004; Erban, 2014). The RMS is denoted by taking the square root of the product of twice the diffusion coefficient of the molecule multiplied by the time step ($RMS = \sqrt{2 * Diffusion\ Coefficent * \Delta t}$) (Erban, 2014; Pryor et al., 2015).

In our simulation at time = t, a molecule has 3 spatial coordinates in the x, y and z plane: x(t), y(t), z(t). At the next increment of time (increased by Δt), the molecule will have positions: x(t+ Δt), y(t+ Δt) and z(t+ Δt). Thus, to calculate the new positions, we apply the following principles (Andrews and Bray, 2004):

$$x(t+\Delta t) = x(t) + \sqrt{2 * Diffusion\ Coefficent * \Delta t} \ * \xi x$$

$$y(t+\Delta t) = y(t) + \sqrt{2 * Diffusion\ Coefficent * \Delta t} \ * \xi y$$

$$z(t+\Delta t) = z(t) + \sqrt{2 * Diffusion\ Coefficent * \Delta t} \ * \xi z$$

where, ξx, ξy and ξz are random numbers obtained from a normal distribution. For the initial set up for the simulations, coordinates for the molecules are generated randomly in Matlab at time 0. These initial coordinates are then fed into the code written in fortran, which is the main programming core and new coordinates are generated according to the above principles at time increments of Δt. In chapter 3, molecules diffuse in the x and y plane only whereas in chapter 4 molecules can also diffuse in the z plane.

1.6.3.3 Confinement zones/domains

The simulation space in our model contains coordinates for confinement zones or domains on the simulation membrane. These domains represent lipid rafts or corrals that might transiently trap receptors and affect their corresponding diffusion and signaling events. The coordinates for these domains are obtained by processing receptor trajectories through the domain reconstruction algorithm (DRA), developed and described in detail by Pryor *et al*., (Pryor et al., 2015). We have implanted the DRA in reconstruction of receptor domains in chapter 4. Briefly, the DRA ranks the receptor trajectories into "slow moving" (receptors confined in a domain) and "fast moving" (receptors not confined in any domain) trajectory points based on their jump sizes over various time frames. The slow moving points are further grouped together by comparing whether their distance from each other is less than the reference distance L. Once a group of slow moving points has been identified, contours can be built around them and coordinates of receptors domains extracted. The coordinates of the domains are then read into the main simulation program by Fortran, along with the initial coordinates of the receptors. Upon diffusion in the simulation membrane, a receptor can find itself trapped in these domains.

1.6.3.4 Simulation boundary conditions

In our simulations, two kinds of simulation boundaries exist: periodic boundary conditions and reflective boundary conditions (Pryor et al., 2015). Periodic boundary conditions exist at the edges of the simulation space (x and y plane). If a molecule is close to the edge of the *in silico* membrane space and the jump is predicted to displace the molecule outside the simulation space, then the jump is divided between the distance travelled before (J1) and the distance travelled after the molecule crosses the edge (J2). In such cases, the molecule travels through the first distance (J1) and completes the rest of the jump distance (J2) by entering the simulation space from the opposite edge of the simulation space. This ensures that the molecules stay inside the simulation space.

Reflective boundary conditions exist at the edges of the membrane domains. Receptors are free to enter their domains, but have to pay a “penalty” to leave their domains. This penalty is in the form of an escape rate probability, which is estimated through SPT data. If a receptor reaches the edge of a domain boundary, the receptor jumps are divided between the distance travelled before (J1) and the distance travelled after the receptor crosses the membrane boundary (J2). A random probability of escaping is generated and if it is not met, then the receptor is simply reflected back into the membrane domain with the second jump distance (J2). If, however, the probability of escape is met then the receptor crosses the membrane boundary.

1.6.3.5 Reaction kinetics

For reaction kinetics, we chose similar principles used in Smoldyn (Smoluchowski dynamics with revisions). Reaction kinetics are based on whether reactions are first order or second order reactions. First order reactions include molecule dissociation, phosphorylation and dephosphorylation. Second order reactions include molecule association (formation of dimers or higher order oligomers). First order reaction probabilities are calculated along using the following method (Pryor et al., 2015):

P(First order reaction) =First order reaction rate *Δt

For second order reactions, we employ the use of a parameter called the binding radius. If two molecules are within the binding radius of each other, then the molecules can form dimers or higher order oligomers upon collision. The binding radius takes into account the reaction on rate, diffusion coefficient and the time step. The unbinding radius for two molecules is 5 times the binding radius and this is done the ensure that there are not too many rebinding events between molecules.

Chapter 2: Predicting chemotherapeutic outcomes in colorectal cancer (CRC) using a mathematical model of drug transport

2.1 NOTES

Data shown in this section, 2, was published in PLOS Computational Biology, titled, "Theory and Experimental Validation of a Spatio-temporal Model of Chemotherapy Transport to Enhance Tumor Cell Kill", on June 2016, August, Volume 12, DOI: DOI:10.1371/journal.pcbi.1004969

Zhihui Wang[1,2,3], Romica Kerketta[4], Yao-Li Chuang[5], Prashant Dogra[4], Joseph D. Butner[6], Terisse A. Brocato[6], Armin Day[4], Rong Xu[7], Haifa Shen[7], Eman Simbawa[8], A. S. AL-Fhaid[8], S. R. Mahmoud[8], Steven A. Curley[9], Mauro Ferrari[7], Eugene J. Koay[10*], Vittorio Cristini1[,2,3,8*]

[1] Department of NanoMedicine and Biomedical Engineering, University of Texas Medical School at Houston, Houston, Texas, United States of America, [2] Brown Foundation Institute of Molecular Medicine, University of Texas Medical School at Houston, Houston, Texas, United States of America, [3] Department of Imaging Physics, University of Texas MD Anderson Cancer Center, Houston, Texas, United States of America, [4] Department of Pathology, University of New Mexico, Albuquerque, New Mexico, United States of America, [5] Department of Mathematics, California State University, Northridge, California, United States of America, [6] Department of Chemical and Biological Engineering and Center for Biomedical Engineering, University of New Mexico, Albuquerque, New Mexico, United States of America, [7] Department of Nanomedicine, Methodist Hospital Research Institute, Houston, Texas, United States of America, [8] Department of Mathematics, Faculty of Science, King Abdulaziz University, Jeddah, Saudi Arabia,[9] Michael E. DeBakey Department of Surgery, Baylor College of Medicine, Houston, Texas, United States of America, [10] Department of Radiation Oncology, University of Texas MD Anderson Cancer Center, Houston, Texas, United States of America

*Correspondence: EKoay@mdanderson.org (EJK); Vittorio.Cristini@uth.tmc.edu (VC)

2.2 ABSTRACT

The physiological barriers in a three dimensional microenvironment of the tumor play a significant role in transporting the drug molecules across the tumor. Herein, we describe a mathematical model of drug transport that takes into account these physiological barriers and based upon physical laws of diffusion, helps to predict chemotherapeutic outcomes in cancer patients. We retrospectively extracted data from histopathological sample of patients that had colorectal cancer (CRC) metastases in the liver. The extracted data were used to populate our mathematical model of drug transport that predicts the fraction of tumor killed based on the following measurable patient specific parameters: radius of blood vessels, blood volume fraction and diffusion penetration length traversed by drugs after extravasation from blood vessels. This predicted value of fraction of tumor killed indicates the effectiveness of the chemotherapy in cancer patients. To validate our model, the parameters radius of blood vessels and diffusion penetration length were derived by fitting our model to histopathological measurements of fraction of tumor killed after chemotherapeutic intervention in human patients with CRC metastatic to the liver (coefficient of determination $R^2 = 0.86$). To test the model feasibility in clinical settings, blood volume fraction values obtained through *in vivo* contrast enhanced computed tomography from the same cohort of patients were used to calculate fraction of tumor killed, where our model was accurately able to predict patient outcomes (average relative error = 24%). Thus, in a clinical setting our model may help in devising patient specific treatment strategies that will help in improving chemotherapeutic outcomes as well as reducing patient expenditure and drug toxicity.

2.3 INTRODUCTION

Predicting chemotherapeutic outcomes in cancer patients is an essential part of overall patient treatment strategy (Cianfrocca and Goldstein, 2004). Based on such predictions, patients who are responding to a certain treatment could be treated accordingly and alternative treatment strategies could be explored for others. This would help in ensuring patient centric care through reduced drug toxicity and optimized utilization of healthcare resources (van 't Veer and Bernards, 2008; Wei et al., 2013).

The extent to which a chemotherapy is effective depends not only on the potency of drugs, but also on the transport of these agents into the tumor sites in effective doses (Kim et al., 2013). This transport of drugs into the tumor site is a complex process taking place over several temporal and spatial scales such as the organ, tissue and cells, and intracellular scales (Kim et al., 2013). At each of these scales, chemotherapeutic drugs face multiple physiological barriers such as variation in the amount of blood supplied to the tumor, an uneven extravasation from the blood vessels, high interstitial fluid pressure (IFP) and large transport distances that they need to overcome in order to reach the tumor site (Jain, 1990; Frieboes et al., 2009). The role of these physiological barriers in impacting the effectiveness of chemotherapies is not well understood. It has been suggested that in order to improve drug penetration into the tumor microenvironment, these physiological barriers must be targeted (Minchinton and Tannock, 2006; Tredan et al., 2007).

Colorectal cancer (CRC) is the third leading cause of cancer related death in the United States (Siegel et al., 2014). Among patients with CRC, about 50% will develop hepatic metastases at some point during their disease (Kanas et al., 2012). Although chemotherapy is routinely administered for the treatment of such metastatic cases, the 5 year survival rates for these patients has been less than 1% and they often have to undergo liver sectioning to increase their chances of survival (Leonard et al., 2005; Gallinger et al., 2013). In such cases, it is important to predict the outcome of chemotherapy so that if a tumor is non-responsive, the treatment can be modified to ensure maximization of net treatment benefits (Cui et al., 2008; Schirin-Sokhan et al., 2012). There is an unmet need for predicting tumor response to chemotherapy in patients with CRC liver metastases as most of the current techniques used for this purpose are ineffective or only applicable after chemotherapy is initiated (Rubbia-Brandt et al., 2007; Glazer et al., 2010; Egger et al., 2013). We have previously developed and validated a steady state diffusion barriers model of drug transport (Pascal et al., 2013b). Herein, we will further revise and validate this model by utilizing a larger set of patient data. Revising and validating our mathematical model will enable more accurate predictions of tumor responses before initiating a chemotherapeutic treatment, hence helping make evidence-based treatment decisions. The purpose of this study is to predict fraction of tumor killed (f_{kill}) based on heterogeneous physiological barriers present in the tumor microenvironment that affect tumor response to chemotherapy using mathematical models of drug transport. We define tumor response to chemotherapy as f_{kill}, which in our mathematical model is calculated through the following patient-specific parameters: radius of blood vessels (rb), blood volume fraction (BVF) and diffusion penetration length (L). We will use histopathological samples of cancer patients to help validate our mathematical model.

2.4 MATERIALS AND METHODS

2.4.1 Steady state diffusion barriers model

The steady state diffusion barriers model consists of differential equations that mathematically describe the process of diffusion and rate of uptake and their parameters. We acquired a biophysical description of the vasculature and tissue architecture present in the tissue surrounding a tumor to form the basis on which we could design our mathematical model that would show the transport of drug within a tumor.

The liver consists of hexagonal shaped lobules and the portal triad (portal artery, portal vein and the bile duct) that transport blood, oxygen, nutrients and bile in the hepatocytes (Rubin E, 2009). We assume that the portal triad can be described effectively as a cylinder and the drug is transported from this cylinder through two physical processes- diffusion and drug uptake- into the surrounding tissue (Figure 2.1). In the model, concentration of local drug within the tumor is obtained by solving an equation that describes diffusion and uptake of drugs into the tumor after transport from the blood vessel has occurred. f_{kill} is dependent on the concentration as well as the effectiveness of the drug at this concentration within the tumor. Hence, f_{kill} is calculated by integrating the effectiveness of the drug around the cylindrical volume which envelopes the blood vessel/drug source. Here, the maximum predicted f_{kill} for each patient is dependent on only 3 parameters- r_b, BVF and L - all of which can be obtained from patient histopathology data. We used the following equation on patients with CRC liver metastases (Pascal et al., 2013b):

$$f_{kill} = 2 \cdot BVF \cdot L \; \frac{\sqrt{BVF} \cdot K_1\,(r_b/L) - K_1\left(r_b/L \cdot \sqrt{BVF}\right)}{\sqrt{BVF} \cdot r_b \cdot K_0\,(r_b/L) \cdot (1 - BVF)} \quad (1)$$

where K_0 and K_1 are modified Bessel functions of the second kind of 0 and 1 orders respectively, f_{kill} is the fraction of tumor killed, r_b is the radius of blood vessel, BVF is the blood volume fraction and L is the diffusion penetration length.

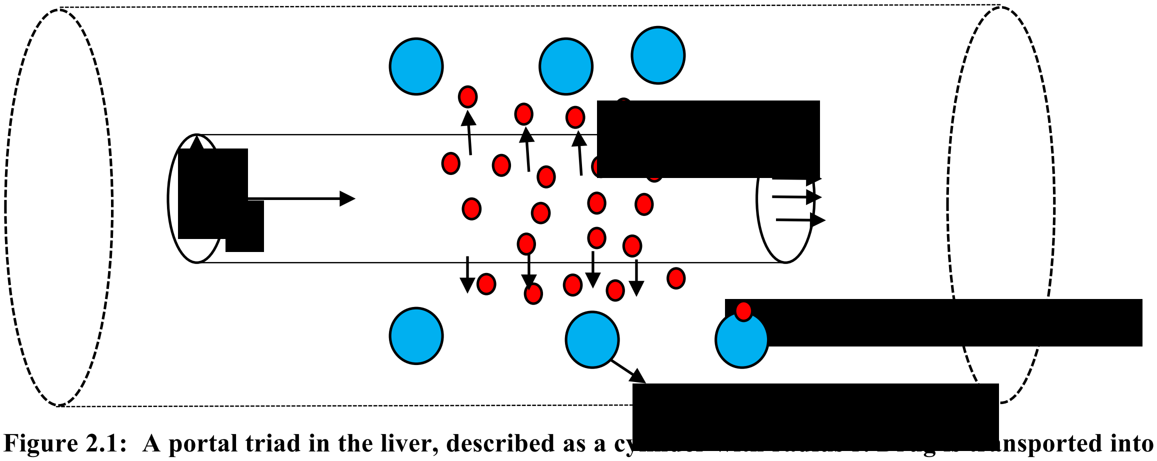


Figure 2.1: A portal triad in the liver, described as a cy[redacted]nsported into the surrounding tissue from the portal triad by diffusion and drug uptake by cells.

2.4.2 Extraction of data from histopathological samples

H&E stained microscopic slides of randomly selected human liver specimen were obtained from a cohort of 27 patients with colorectal cancer (CRC) metastatic to liver at the MD Anderson Cancer Center (MDACC). Six patients were not included in the final analysis because they lacked dead tumor tissue. Based on the assumption that histologic sections are isotropic, fraction of tumor killed f_{kill} was directly measured as fraction of tumor killed in histologic assessments along with measurements of radius of blood vessels r_{b}, and blood volume fraction (BVF) for each patient (20 slides per patient). Measurements were manually performed using GNU Image Manipulation (GIMP) and illustrations of measurement of r_{b} are in Figure 2.2. In order to calculate the fraction of dead tumor area, dead areas of tumor were colored red, live areas of tumor were colored blue and the portions that were not tumor i.e. normal were colored green. Fraction of dead tumor area was set as:

f_{kill} = # of red pixels / (# of red pixels+ # of blue pixels+ # of green of pixels)

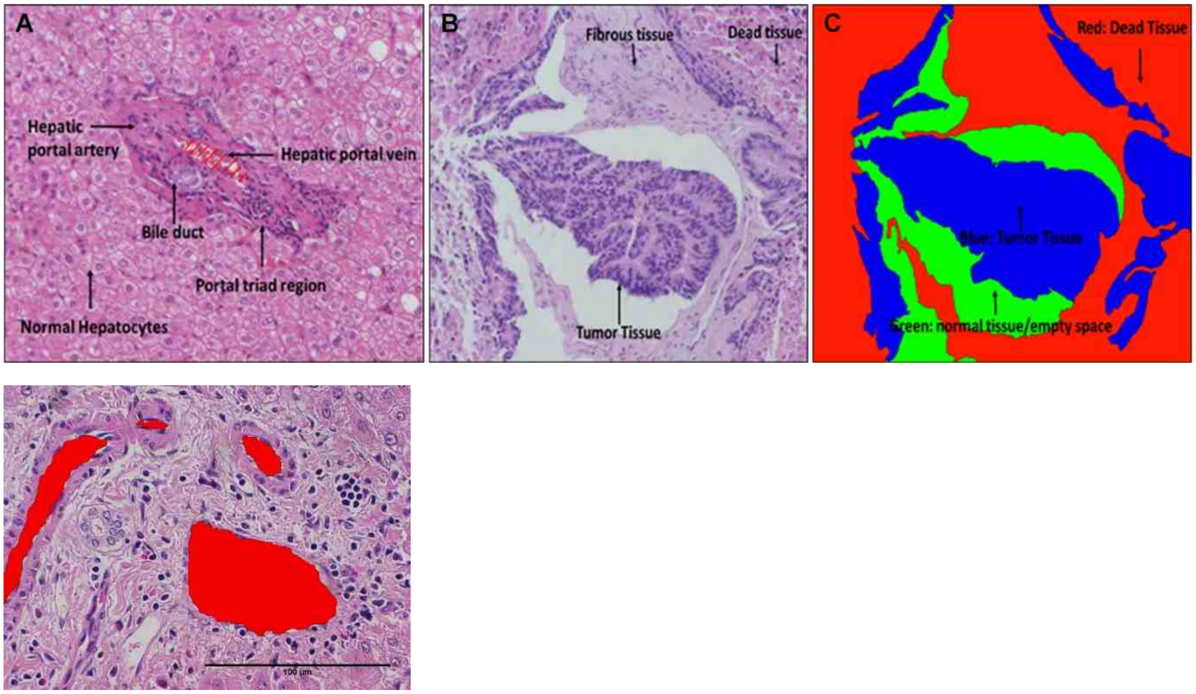


Figure 2.2: Example of measurements from histopathological specimens of patient data. (A) Example of portal triad with blood vessel measurements. (B) Example of a histologic section. (C) Segmentation of the histologic section B for calculation of the fraction of dead tumor area: dead tumor (red); live tumor (blue); no tumor (green). (D) Segmentation of a histologic section for calculation of blood volume fraction: blood vessels (red).

We compared f_{kill} values between pathologist's measurements and our measurement of f_{kill}. A cumulative Distribution Frequency (CDF) graph was generated to compare the f_{kill} values (Figure 2.3A). From the graph we concluded that the pathologist f_{kill} values were shifted by some value to the right from our measurements because when pathologists take their measurements of f_{kill}, they don't take into account whether normal tissue is destroyed and only see what is dead and live tumor. This needed to be corrected because our f_{kill} values should correlate with what the pathologist measure in real life. That value by which our f_{kill} values needed to be shifted by was calculated by taking the difference between the average of the f_{kill} measured by us and average of the f_{kill} measured by the pathologist and that value came to be 0.108411284. Another CDF graph was generated to compare our new f_{kill}, with the added value (Figure 2.3 B). These new f_{kill} values were then further used to do the fitting in Mathematica.

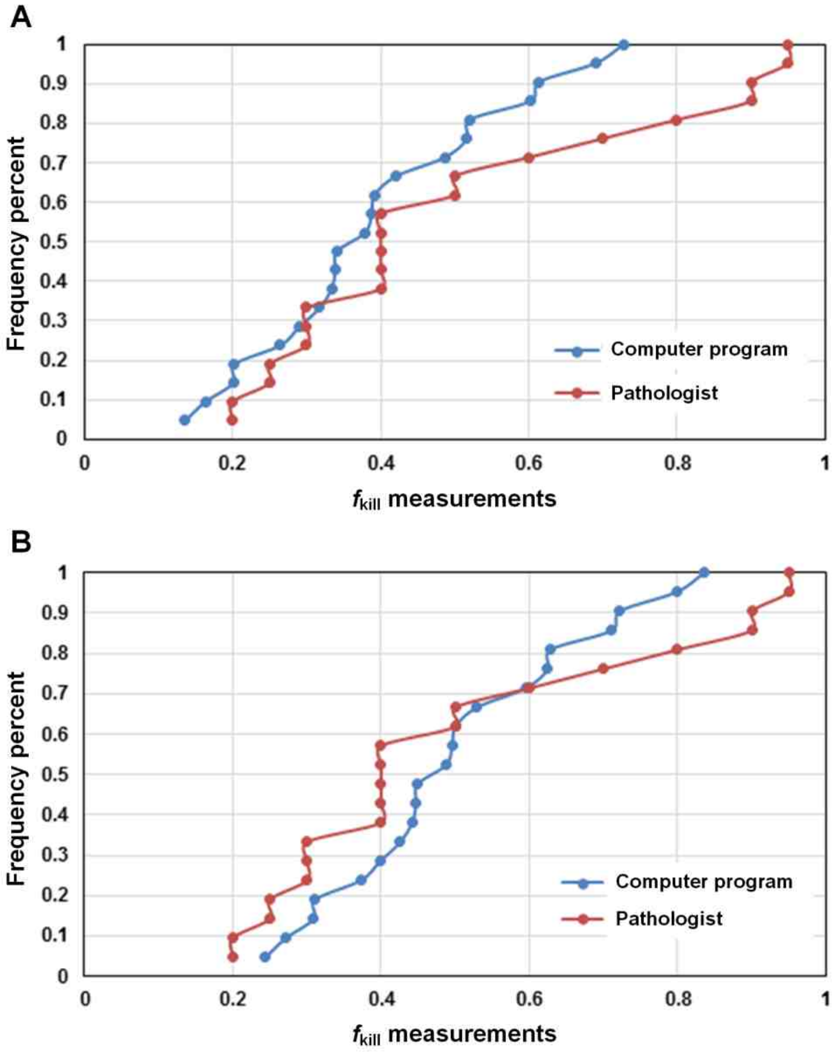


Figure 2.3: Cumulative Distribution Frequency (CDF) graphs. (A) Cumulative Distribution Frequency (CDF) graph of f_{kill} values comparing measurements from pathologist versus from computer program (GIMP). (B) Corrected Cumulative Distribution Frequency (CDF) graph of f_{kill} values.

2.4.3 Test of model predictivity

To test the model predictivity based on three parameters r_{b}, BVF, and L, the following mathematical equation was applied to this cohort of patients:

$$f_{\mathrm{kill}} = \mathrm{BVF} \cdot L \cdot \left[\frac{2 \cdot \sqrt{\mathrm{BVF}} \cdot K_1(r_{\mathrm{b}}/L) - 2 \cdot K_1(r_{\mathrm{b}}/(L \cdot \sqrt{\mathrm{BVF}}))}{\sqrt{\mathrm{BVF}} \cdot r_{\mathrm{b}} \cdot K_0(r_{\mathrm{b}}/L) \cdot (1 - \mathrm{BVF})} \right] \qquad \textbf{[1}$$

r_{b} and L were obtained from the regression analysis of the MDACC cohort. The model was fitted to the measured f_{kill} values and their corresponding BVFs and a graph was generated by comparing the predictions of Eq.1 to the direct measurements of kill (Figure 2.4). Least-squares fitting of Eq. 1 was performed using Mathematica routine “NonlinearModelFit” to the kill fraction and BVF measured in liver metastasis in the MDACC patient cohort. This resulted in estimates of parameters r_{b} and L (diffusion penetration length), which produced the best fit.

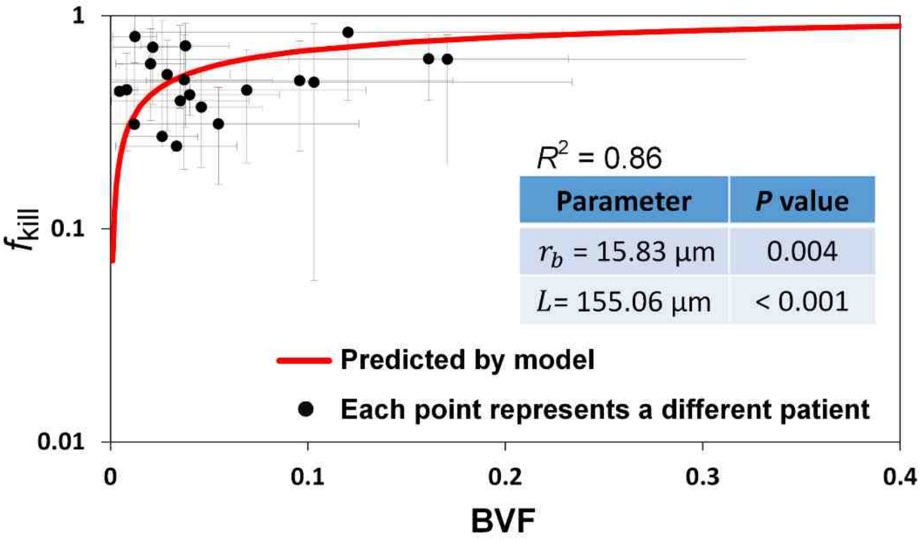


Figure 2.4 : Fitting the model to patient data demonstrates biological accuracy of the functional form of Eq. 1. Symbols: measurements with standard deviations from histopathology images of 21 patients with CRC metastatic to liver. Standard deviations calculated by measuring variability in patient data. Red: least-square fit of Eq. 1 to the data (R2 = 0.86).

2.4.4 Prospective application of the model based on pretreatment CT scans

Contrast CT scans performed according to standard clinical protocols were acquired prior to chemotherapy on 18 scans performed according to standard clinical protocols were acquired prior to chemotherapy on 18 patients at MDACC according to institutional review board-approved protocols. The simple average of three Hounsfield Unit (HU) measurements in representative areas within the entire tumor was calculated at the each phase of the test for each patient i.e., a late arterial phase (30-35 s after start of contrast injection), a portal venous phase (50-55 s), and a delay phase (minutes, variable timing).

The pre-treatment CT measurements (HU) at the arterial phase were found by linear regression analysis to correlate to the measurements of BVF (blood volume fraction) performed from post-treatment histology (Figure 2.5):

$$\mathrm{BVF} = 0.00091672\ \mathrm{CT\ (HU)}, \qquad \textbf{[2]}$$

with coefficient of determination R^2 = 0.63 (Devore, 2011), *p*-value = 0.00008, from Mathematica routine “LinearModelFit” (Wolfram Research, 2008) and GraphPad Prism (GraphPad Software, 2007). CT measurement error of 25% was estimated from corresponding data of contrast enhancement in the aorta, and thus represents variability in physiology and contrast dosing in CT protocol across patients. Even with a limited number of subjects, the statistical significance (*p*-value = 0.00008) is expected since CT measurements reflect perfusion of tissue, which relies on the volume fraction of blood vessels. Analysis using the portal-venous measurements produced similar results. We multiplied the regression coefficient with the CT scan data to find the individual corresponding *predicted* BVF. The BVF, r_b, and L values were then inserted into Eq. 1 to obtain the *predicted* f_{kill} values (Figure 2.6). An error of 25% was found in the CT measurements and this was estimated by comparing the corresponding measurements from contrast enhancement of aorta (standard deviation in aorta/average of aorta). Thus error bars for the model predictions based on the CT scan data were calculated using this 25% error estimated in CT measurements. The upper and lower limit came from ± 25% of the *predicted* BVFs and this gave rise to an upper and lower limit for f_{kill}, which were then used in the calculation of standard error in *predicted* f_{kill} . The average relative error between the model prediction f_{kill} (P) and the measured kill value f_{kill} (M) was calculated as: $<(f_{\text{kill}}(\text{P})-f_{\text{kill}}(\text{M}))/f_{\text{kill}}(\text{M})>$. Outliers more than 2 SD away were removed from the calculation of the relative error.

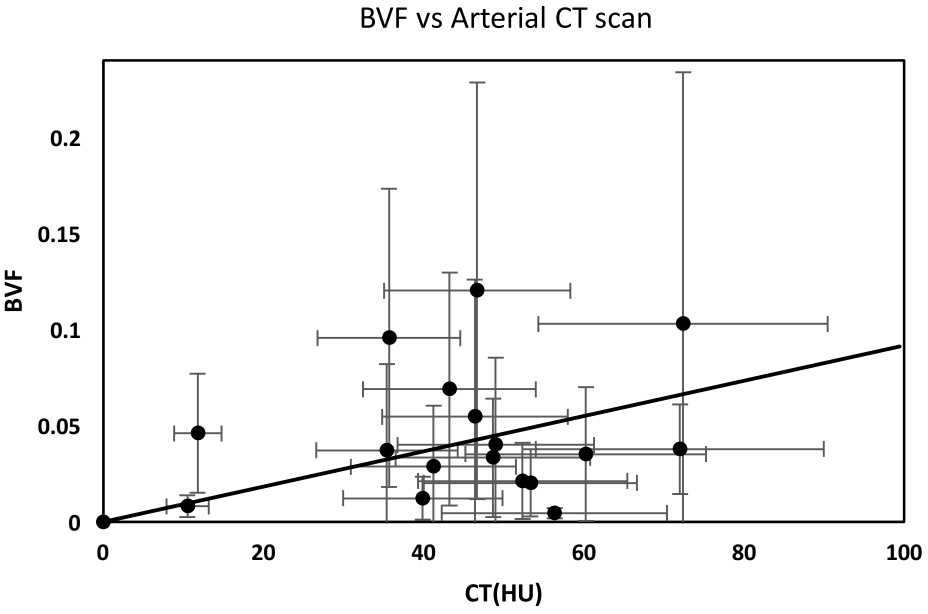


Figure 2.5: Calculation of regression coefficient between CT scan data and *BVF*. Regression coefficient = .00091672.

2.5 RESULTS

2.5.1 Fitting the mathematical model to patient data by a regression Analysis identifies biologically realistic parameter values

The regression analysis resulted in estimates of the two parameters: r_b and L from the MDACC cohort of patients. Each of these estimated values is consistent with measurements from human anatomy. The radius of the blood vessels in the portal triad, $r_b \approx 15.8$ μm obtained from this fitting was consistent with published data (Wiedeman, 1963; Muraca, 1994) and with our histopathology (Figure 2.2). The diffusion penetration length from regression analysis is $L \approx 151$ μm. Using *Mathematica* (Wolfram Research, 2008), statistically significant *p*-values were obtained for both r_b and L. (Figure 2.4, inset).

2.5.2 Prospective application of the mathematical model *in vivo*

To establish whether our model could predict chemotherapy outcome based only on standard pre-treatment contrast-CT imaging, we carried out the following series of steps. First we performed an analysis on the histopathology from post-treatment specimens on the MDACC cohort of patients. This validated the predictive power of Eq. 1 specifically for this cohort of patients (Figure 2.4, red curve: $R^2 = 0.86$). Here we used the same value of diffusion penetration distance L and portal radius r_b obtained from regression analysis on all patients, which again point to uniformity of these parameters across patients thus generating the hypothesis that future clinical translation would primarily rely on patient-specific calculation of the parameter BVF. To test this hypothesis we calculated a linear correlation constant for histopathology BVF and contrast-CT Hounsfield units, which allowed us to obtain a BVF value from the contrast enhancement of the CT images for each individual (Eq. 2). Inputting this value into Eq. 1 produced accurate kill-ratio predictions (Figure 2.6, open circles) that compared well to the actual measurements from histopathology post-treatment (Figure 2.6, filled circles), with an average relative error of the predicted fraction killed of ≈ 24% (*Methods*).

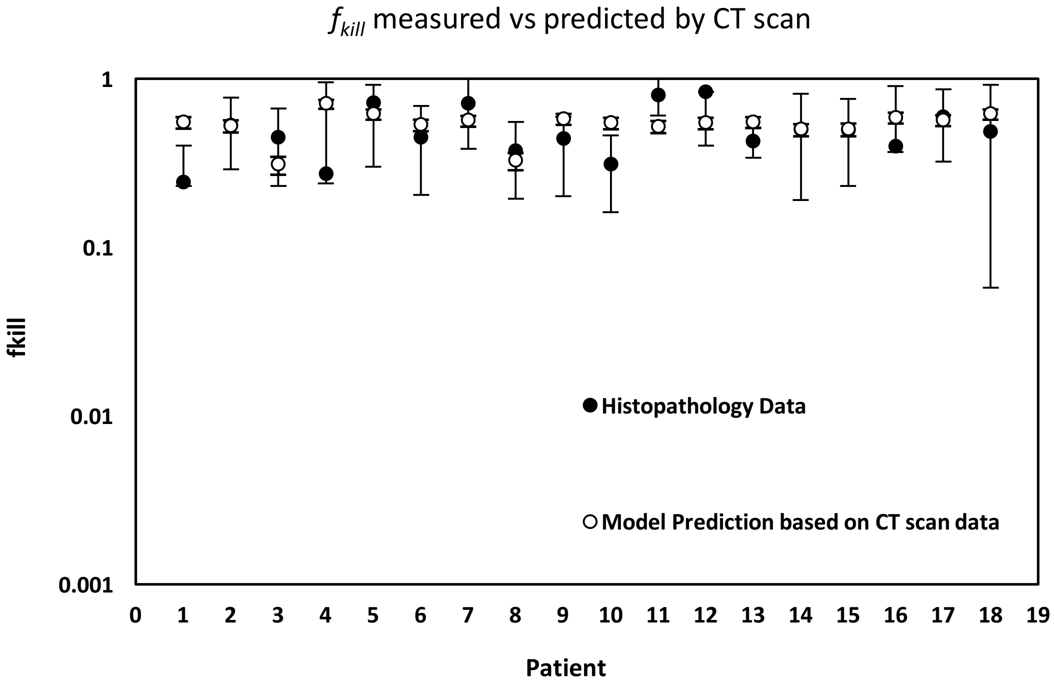


Figure 2.6: Prospective, patient-specific model predictions match outcomes of fraction of tumor killed by chemotherapy in the MDACC cohort of patients with CRC metastatic to liver. Predictions of Eq. 1 using BVF parameter calculated from pre-treatment contrast-CT perfusion measurements (open circles). Multiple measurements from histopathology post-treatment per patient indicated by standard deviation (filled circles). Model input parameters r_b (radii of blood vessels in liver portal triad) and L (drug diffusion penetration distance) from Fig. 1.4

2.6 DISCUSSION

The mathematical model presented here was able to predict fraction of tumor killed in patients with CRC metastatic to liver quite accurately. The average relative error between model predictions and fraction of tumor killed was only 24%, with patient parameters obtained directly from CT scan data. Given the pressing need for potential and robust biomarkers to predict chemotherapy outcomes in patients, our results here provide an unprecedented approach of mitigating this need at a much earlier stage in cancer treatment than some of the other techniques. Imaging techniques such as CT scan, magnetic resonance imaging (MRI) and positron emission tomography (PET) are used to detect changes in tumor in response to chemotherapeutic drugs, however, these changes are quantifiable only about halfway through the treatment, and the patient might have already by this time been exposed to a significant amount of toxicity from the drugs and incurred patient related expenses (Koh and Padhani, 2006; Cui et al., 2008). Other techniques like using Apparent Diffusion Coefficient (ADC) values obtained from diffusion weighted-MRI have the potential to gauge tumor response at an earlier stage (Koh and Padhani, 2006; Vandecaveye et al., 2006; Cui et al., 2008) and this technique has already been used to assess tumor response to chemotherapy in hepatic metastases (Theilmann et al., 2004; Koh et al., 2007; Cui et al., 2008). Our approach of using "mathematical pathology" not only helps to predict the tumor response to drugs before the start of any treatment, thus lessening the cost and toxicity that the patients might be exposed to, but also provides an understanding of the physiological barriers that are responsible for the resistance of drugs to chemotherapy. The model helps to highlight the biological barriers as important players that hamper drug delivery and ones that need a further thorough investigation along with the genetic and cellular reasons for chemotherapy resistance. This model has shown that patient specific physiological features such as blood vessel radius and blood volume fraction play significant roles in determining the amount of drug supplied to the tumor and in a clinical setting these parameters can easily be measured through CT scans and used in the mathematical model for prediction of tumor killed. Thus, patient specific strategies can be developed and patient dosage and timing can be primed for optimal results for each individual. This model can be used alone in the clinical setting to predict the fraction of tumor killed or used with other methods of predicting tumor size such as using the ADC from diffusion weighted- MRI. This study also lays future groundwork to evaluate the effects of other drug carriers such as nanoparticles on drug delivery and tumor size since nanoparticles might prove to be more effective carriers of drugs than just free drug alone. Additional layers of complexity involving other factors or physiological barriers can be added to the model, as the research on physiological barriers continues, and this will result in even more accurate predictions of the tumor response to chemotherapy in the future.

2.7 AUTHOR CONTRIBUTIONS

Designed the research: ZW EJK VC. Conceived the perfusion-based drug resistance hypothesis and designed the mathematical model: VC. Developed the mathematical model: ZW YLC. Per-formed model analysis: ZW RK YLC PD JDB TAB AD ES ASAF SRM. Performed patient data analysis: RK YLC PD JDB TAB AD. Designed and performed the PSP-based experiments in mice: RX HS MF EJK. Obtained histopathological material and performed pathological diag- nostics: SAC EJK. Identified subjects: SAC. Obtained CT scans: EJK. Wrote the paper: ZW RK YLC EJK VC.

2.8 FUNDING

This work has been supported in part by the National Science Foundation (NSF) Grant DMS- 1562068 (ZW, VC), the National Institutes of Health (NIH) Grant 1U01CA196403 (ZW, EJK, VC), 1U54CA143837 (SAC, MF, VC), 1U54CA151668 (MF, VC), the University of Texas System STARS Award (VC), the Methodist Hospital Research Institute (HS, MF, EJK, VC), and the New Mexico Cancer Nanoscience and Microsystems Training Center (CNTC) Graduate Student Fellowship (RK, TAB). This work was also funded in part by the Deanship of Scientific Research (DSR), King Abdulaziz University, under Grant no. (HiCi/54-130- 35) (ES, ASAF, SRM, VC). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

CHAPTER 3: EFFECT OF SPATIAL INHOMOGENEITIES ON THE MEMBRANE SURFACE ON RECEPTOR DIMERIZATION AND SIGNAL INITIATION

3.1 NOTES

The work shown in this section, 3, was published in Frontiers in Cell and Developmental Biology on August 2016, August, Volume 4, Article 81, Pages 1-13, DOI: 10.3389/fcell.2016.00081

Romica Kerketta [1], Ádám M. Halász [2], Mara P. Steinkamp [1, 3], Bridget S. Wilson [1, 3] and Jeremy S. Edwards [3, 4, 5, 6*]

[1] Department of Pathology, University of New Mexico Health Sciences Center, Albuquerque, NM, USA, [2] Department of Mathematics and Mary Babb Randolph Cancer Center, West Virginia University, Morgantown, WV, USA, [3] Cancer Center, University of New Mexico Health Sciences Center, Albuquerque, NM, USA, [4] Department of Chemical and Biological Engineering, University of New Mexico, Albuquerque, NM, USA, [5] Department of Chemistry and Chemical Biology, University of New Mexico, Albuquerque, NM, USA, [6] Department of Molecular Genetics and Microbiology, University of New Mexico, Albuquerque, NM, USA.

*Correspondence: Jeremy S. Edwards jsedwards@salud.unm.edu

3.2 ABSTRACT

Important signal transduction pathways originate on the plasma membrane, where microdomains may transiently entrap diffusing receptors. This results in a non-random distribution of receptors even in the resting state, which can be visualized as “clusters” by high resolution imaging methods. Here, we explore how spatial in-homogeneities in the plasma membrane might influence the dimerization and phosphorylation status of ErbB2 and ErbB3, two receptor tyrosine kinases that preferentially heterodimerize and are often co-expressed in cancer. This theoretical study is based upon spatial stochastic simulations of the two-dimensional membrane landscape, where variables include differential distributions and overlap of transient confinement zones (“domains”) for the two receptor species. The *in silico* model is parameterized and validated using data from single particle tracking experiments. We report key differences in signaling output based on the degree of overlap between domains and the relative retention of receptors in such domains, expressed as escape probability. Results predict that a high overlap of domains, which favors transient co-confinement of both receptor species, will enhance the rate of hetero-interactions. Where domains do not overlap, simulations confirm expectations that homo-interactions are favored. Since ErbB3 is uniquely dependent on ErbB2 interactions for activation of its catalytic activity, variations in domain overlap or escape probability markedly alter the predicted patterns and time course of ErbB3 and ErbB2 phosphorylation. Taken together, these results implicate membrane domain organization as an important modulator of signal initiation, motivating the design of novel experimental approaches to measure these important parameters across a wider range of receptor systems.

3.3 KEYWORDS

Spatial stochastic modeling, membrane domains, ErbB receptors, ErbB2, ErbB3

3.4 INTRODUCTION

The plasma membrane is the initiation site for signaling pathways that govern cell differentiation, proliferation and survival (Groves and Kuriyan, 2010; Radhakrishnan et al., 2012). The membrane provides a platform for the reversible binding of ligands to receptors, initiating critical processes such as dimerization, activation of catalytic activity and recruitment of binding partners (Groves and Kuriyan, 2010). Given its importance in cell signaling, the structure and composition of membranes have been probed by many different groups. Singer and Nicholson, in their landmark paper of the fluid mosaic model, proposed membranes to be largely homogenous with randomly distributed mixtures of integral membrane proteins and lipids (Singer and Nicolson, 1972). However, the authors also showed electron microscopy images of major histocompatibility antigen "patches," providing early evidence for membrane organization. Since then, considerable evidence has accumulated showing that membrane proteins and lipids can be transiently confined in specific domains (Kaizuka et al., 2007; Chung et al., 2010; Treanor et al., 2010; Radhakrishnan et al., 2012; Goñi, 2014). The anomalous diffusion of membrane constituents, observed through single molecule tracking methods (Fujiwara et al., 2002), is likely due, at least in part, to their transient entrapments within heterogeneous domains (Marguet et al., 2006). Multiple theories exist to explain the richness of the plasma membrane topography, including lipid rafts which are enriched in unsaturated fatty acids and cholesterol (Pike, 2003), corrals formed by the actin cortical cytoskeleton network

(Jaqaman et al., 2011; Kalay, 2012; Cambi and Lidke, 2015) and protein islands (Lillemeier et al., 2006). Even very short periods of confinement within domains give rise to lateral heterogeneity and an uneven distribution of proteins on the membrane surface that can be captured in "snap-shot" images by electron microscopy of membrane rip-flips (Wilson et al., 2000; Prior et al., 2001; Andrews et al., 2009). More recently, super-resolution microscopy methods have also been employed to document the clustering of membrane proteins (van den Dries et al., 2013; Itano et al., 2014). The exchange of proteins between domains is highly variable, ranging from very low exchange rates observed in yeast membranes (Spira et al., 2012) to very rapid exchanges described for the EGFR in mammalian cell membranes (Low-Nam et al., 2011).

Many important receptors exhibit varying degrees of clustering prior to ligand engagement, including members of the EGFR/ErbB family (Nagy et al., 2002; Yang et al., 2007) and the ITAM-bearing immunoreceptors (FcεRI, BCR, TCR) (Pike, 2003; Lillemeier et al., 2006; Andrews et al., 2009; Tolar et al., 2009; Treanor et al., 2010; Dinic et al., 2015). Experimental evidence has suggested that membrane domains can both enhance and inhibit signaling in different settings (Marmor and Julius, 2001; Miura et al., 2001; Douglass and Vale, 2005; Allen et al., 2007; Bénéteau et al., 2008; Ganguly et al., 2008). Computational studies have also supported the concept that membrane organization has cell and receptor-specific outcomes (Lim and Yin, 2005; Hsieh et al., 2008; Costa et al., 2011; Abel et al., 2012; Kalay et al., 2012). For example, the formation of different signaling clusters has been proposed to support distinct TCR signaling patterns (Singleton et al., 2009). Vale and colleagues recently demonstrated in model membranes that phase separation of signaling partners can create distinct signaling compartments (Su et al., 2016). Members of the ErbB family of receptor tyrosine kinases have been shown to have distinct distribution patterns on cancer cell membranes (Yang et al., 2007; Steinkamp et al., 2014), leading to computational studies from our group that predict the impact of critical variables such as receptor co-expression, density and dimer off-rates (Hsieh et al., 2008; Pryor et al., 2013, 2015).

Deterministic models based upon Ordinary Differential Equations (ODEs) are not well suited to explore spatial aspects of signaling, since they assume molecules in a system are well mixed. Stochastic modeling approaches offer greater flexibility to consider effects of membrane topography, receptor clustering and diffusion dynamics on signaling events (Mayawala et al., 2006; Nicolau et al., 2006; Hsieh et al., 2008; Costa et al., 2009; Chaudhuri et al., 2011). These versatile mathematical models provide a platform for rapid exploration of key factors that are difficult to vary (and measure) experimentally. In this study, we take advantage of this powerful approach to consider the effect of two parameters, membrane domain overlap and domain retention, on ErbB3 and ErbB2 homo-and heterodimerization. Our group previously evaluated the domain occupancy and distribution of ErbB2 and ErbB3 stably expressed as recombinant proteins in Chinese Hamster Ovary (CHO) cells (Steinkamp et al., 2014; Pryor et al., 2015). Analysis of dual-color single particle tracking data, which permitted independent observations of each species, indicated that domains confining the two ErbB receptors were only partially overlapping in the CHO cell membrane (Pryor et al., 2015). We then built a spatial stochastic model based upon this distribution, as well as experimentally measured values for dimer off-rates, kinase/phosphatase activity and receptor diffusion (Pryor et al., 2015). However, we speculate that the degree to which there is differential segregation of these two closely related receptors will vary widely as a property of cell type, because of dissimilar receptor ratios, density, cytoskeletal features, membrane composition and on-going signal transduction from other cell surface receptors triggered by circulating or local ligands. In this paper, we focus on two specific parameters that affect the degree to which ErbB2 and ErbB3 experience periods of co-confinement: domain overlap and retention, where the latter is expressed as a function of escape probability.

3.5 MATERIALS AND METHODS

3.5.1 Spatial stochastic model for ErbB2 and ErbB3 homo- and hetero-dimerization

3.5.1.1 Reactions

The spatial stochastic model of ErbB2 and ErbB3 interactions was described previously (Pryor et al., 2015). Briefly, the model includes two members of the EGFR family, ErbB2 and ErbB3, which diffuse within the simulation space and interact with each other.

The following reactions are accounted for in the model:

(i) Dimerization: Homo- and heterodimerization of ErbB2 and ErbB3 receptors.

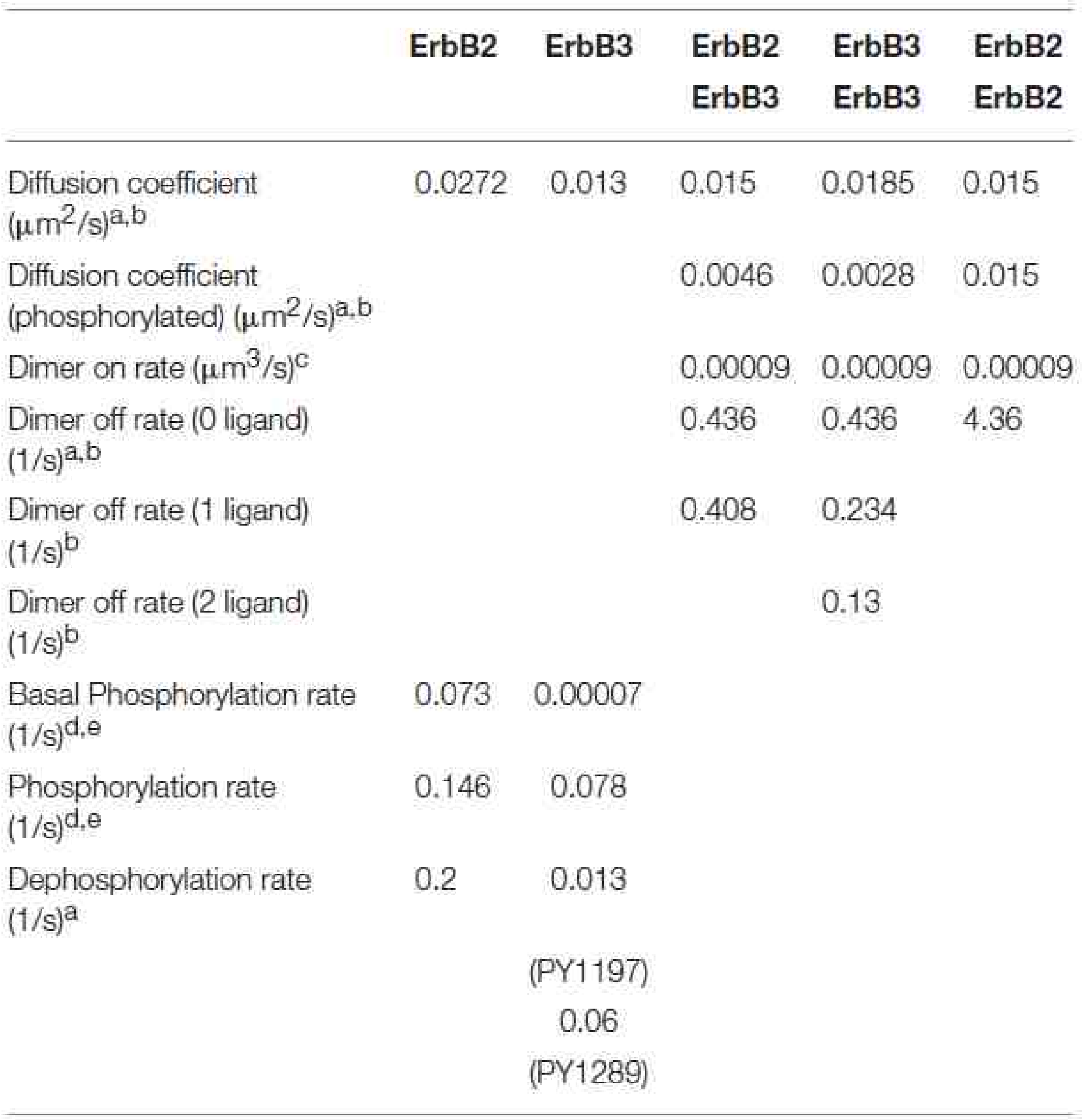
(ii) Phosphorylation: Receptors are phosphorylated through intrinsic phosphorylation rates.

(iii) Dephosphorylation: Receptors are dephosphorylated through experimentally determined dephosphorylation rates.

(iv) Dissociation: Dimer dissociation occurs through experimentally determined dimer off rates.

We assume that the dimerization of receptors occurs through the interaction of the dimerization arms on the extracellular domain of receptors. In the absence of ligand, the ErbB3 extracellular domain fluxes from a closed (tethered) to an open (dimer-competent) conformation. The open conformation of ErbB3 is stabilized by ligand binding (Pryor et al., 2015). Unliganded ErbB3 is assumed to be predominately closed (99.99% closed). At any given time step, there is a 10^{-4} probability for unoccupied ErbB3 receptors to assume the upright dimer-competent state while all ligand-bound ErbB3 monomers are dimer-competent (Hsieh et al., 2008). ErbB3 ligand concentrations vary in the simulations as described in the legends. ErbB2 receptors are assumed to be in open conformation and dimerization competent (Cho et al., 2003; Garrett et al., 2003). In the model, ErbB2 has a single representative tyrosine phosphorylation site based on uniform dephosphorylation kinetics over two tested phosphorylation sites (Pryor et al., 2015). ErbB3 has two representative phosphorylation sites based upon (Y1289; Y1197). Table 1 lists the reaction parameters used in our model including receptor dimerization, phosphorylation/dephosphorylation, and receptor dissociation as previously described (Pryor et al., 2015). For receptor phosphorylation events, the model takes into consideration the asymmetric orientation of kinase domains which occurs during ErbB receptor activation (Ward and Leahy, 2015). Reactions are governed by binding radii estimated using SMOLDYN, a software application that takes into consideration receptor on-rates, diffusion coefficients and simulation time steps to construct a binding radius (Andrews and Bray, 2004). An unbinding radius of 5 times the binding radius was used to decrease rebinding events.

TABLE 1: Model parameters of receptor monomers and dimers



	ErbB2	ErbB3	ErbB2 ErbB3	ErbB3 ErbB3	ErbB2 ErbB2
Diffusion coefficient ($\mu m^2/s$)[a,b]	0.0272	0.013	0.015	0.0185	0.015
Diffusion coefficient (phosphorylated) ($\mu m^2/s$)[a,b]			0.0046	0.0028	0.015
Dimer on rate ($\mu m^3/s$)[c]			0.00009	0.00009	0.00009
Dimer off rate (0 ligand) (1/s)[a,b]			0.436	0.436	4.36
Dimer off rate (1 ligand) (1/s)[b]			0.408	0.234	
Dimer off rate (2 ligand) (1/s)[b]				0.13	
Basal Phosphorylation rate (1/s)[d,e]	0.073	0.00007			
Phosphorylation rate (1/s)[d,e]	0.146	0.078			
Dephosphorylation rate (1/s)[a]	0.2	0.013 (PY1197) 0.06 (PY1289)			

[a] Pryor et al. (2015).
[b] Steinkamp et al. (2014).
[c] Pryor et al. (2013).
[d] Kleiman et al. (2011).
[e] Shi et al. (2010).

3.5.1.2 Simulation landscape

The simulation landscape contains receptor specific domains (Figure 3.1A) and receptors can diffuse across domains and domain-free areas. An exit penalty limits receptor escape from the domains. Figure 3.1A depicts domains that were identified in previous work (Pryor et al., 2015). Represented by a rectangular box measuring 0.1995 μm2 in area (Figure 3.1A), the space contains 5 ErbB2 and 9 ErbB3 receptor domains. These domains were derived from domain analysis of two-color single particle tracking data where ErbB3 was labeled with HRG-conjugated quantum dot (QD) and HA-tagged ErbB2 was labeled with anti-HA Fab conjugated QD (Pryor et al., 2015). The total ErbB2 domain area is 0.0502 μm2; the total ErbB3 domain area is 0.0274 μm2.The free area outside the domains is 0.1219 μm2. We then created three distinct domain overlap conditions for comparison:

(i) 100% overlap: 100% of the ErbB3 domain area is overlapping with the ErbB2 domain area. This resulted in complete mixing of ErbB3 and ErbB2 domains (Figure 3.1D).

(ii) 50% overlap: 50% of the ErbB3 domain area is overlapping with ErbB2 domain area. This resulted in partial overlapping of ErbB3 and ErbB2 domains (Figure 3.1C).

(iii) 0% overlap: 0% of the ErbB3 domain area is overlapping with the ErbB2 domain area. This resulted in complete separation of ErbB3 and ErbB2 domains (Figure 3.1B).

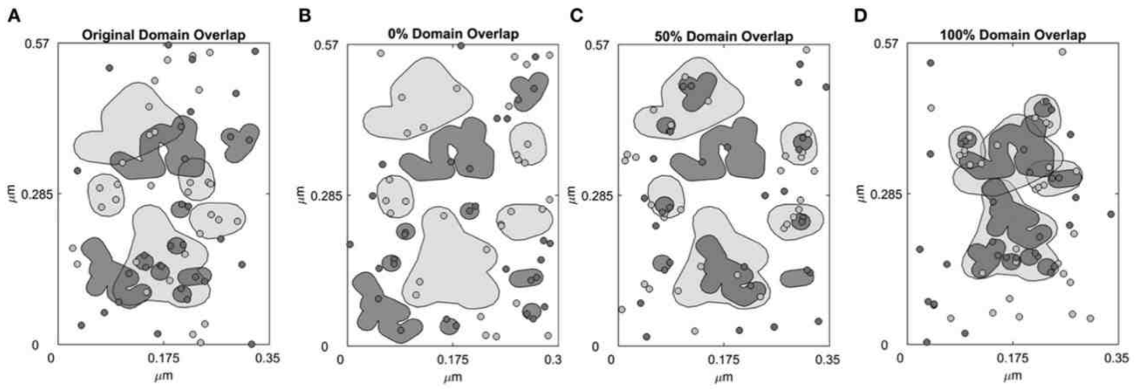


Figure 3.1: Four domain configurations of the simulation space. Simulation space was partitioned into receptor-specific domains with defined domain overlaps. (A) A simulation space that mimics the domain properties of CHO cells overexpressing ErbB2 and ErbB3 based on domain analysis of SPT data. ErbB2 (light gray, shaded) and ErbB3 (dark gray, shaded) membrane domains overlap by 42.4%. ErbB2 receptors (light gray, circled) and ErbB3 receptors (dark gray, circled) are randomly distributed within their own domains as well as outside the domains (white region). (B–D) Domains were rearranged to create a simulation space where the ErbB2 and ErbB3 domains are completely non-overlapping (0% overlap, B), partially overlapping (50% overlap, C) or completely overlapping (100% overlap, D). In the initial configuration, ErbB2 and ErbB3 receptors were positioned to randomly occupy their respective domains.

3.5.1.3 Number and density of receptors

The model was populated with 50,000 ErbB2 and 50,000 ErbB3 receptors/cell. Since the total area of a cell is 314.16 μm2 (with a diameter of 10 μm), this translates into a receptor density of ~159 receptors/μm2 for each receptor. Adjusted for a simulation area of 0.1995 μm2, the total number of receptors is 31 of each receptor species.

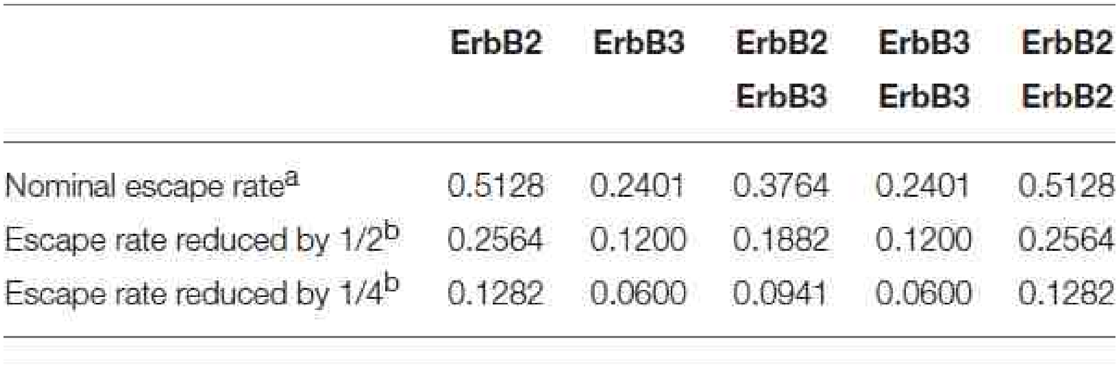
3.5.1.4 Receptor diffusion

Receptor diffusion occurs in the two dimensional membrane simulation space (x and y direction) through Brownian motion. Receptor jumps in these two directions are calculated using diffusion coefficients generated from SPT data and normally distributed random numbers.

3.5.1.5 Boundary conditions

As in Pryor *et al*. (2015) and Pryor *et al*. (2013), the periodic boundary condition is applied to the edges of the simulation space. If a receptor jump takes the receptor across the edge of the simulation space, the jump distance is divided between the distances covered before and after the boundary is crossed. The receptor then traverses the distance to the boundary and the remaining distance is calculated from the opposite edge of the simulation space. Hence, the receptor "re-enters" the simulation space from the opposite boundary. Reflective boundary conditions are applied when a receptor reaches the edge of a membrane domain. Like the periodic boundary conditions, the jump distance is divided between the distances covered before and after reaching the boundary. A probability for crossing/escaping from the membrane boundary is calculated and if the probability of escaping is not met, then the receptor hits the boundary and is deflected back into the domain. If the probability of escape is met, then the receptor continues across the boundary. Escape rates in Pryor et al. (2015) were estimated by parameter fitting to the ratio of domain-confined receptors experimentally measured in CHO cell membranes; this rate is a key variable of the present study (Table 2).

TABLE 2: Escape rates of receptor monomers and dimers



	ErbB2	ErbB3	ErbB2 ErbB3	ErbB3 ErbB3	ErbB2 ErbB2
Nominal escape rate[a]	0.5128	0.2401	0.3764	0.2401	0.5128
Escape rate reduced by 1/2[b]	0.2564	0.1200	0.1882	0.1200	0.2564
Escape rate reduced by 1/4[b]	0.1282	0.0600	0.0941	0.0600	0.1282

[a] Pryor et al. (2015).
[b] Simulation data in this paper.

3.5.1.6 Simulation code

Input files containing the initial simulation space, receptor locations and ligand concentrations are generated in Matlab. These files are then accessed by a program written in Fortran, which simulates brownian diffusion and molecular interactions between the two receptors. At the end of the simulations, all output files are processed in Matlab for analysis of results. Code is available upon request.

3.6 RESULTS

3.6.1 Domain overlap affects the frequency of hetero-interactions and receptor phosphorylation events

It is unknown to what extent different receptors share the same membrane domains, how fluid these domains are over time, and whether activation of receptors alter domain overlap. Therefore, we explored these possibilities through simulations, reporting results as changes in homo- and hetero-dimerization and phosphorylation status. Unlike prior work fit to cells overexpressing ErbB family members (Pryor et al., 2013, 2015), we used receptor densities within the range of expression values expected for normal cells (50,000 receptors/cell). The simulation landscape included either no domains or ErbB2 and ErbB3-specific domains with partial, full or no overlap (Figure 3.1).

The rapid cycling of ErbB3 receptors through different states is illustrated in Figure 3.2, where simulations were initially performed in a landscape lacking domains. Here, ligand-bound ErbB3 freely diffuse, encountering other ErbB3 or ErbB2 monomers with no barriers imposed. They constantly cycle through homodimer (red), heterodimer (orange) and monomer (white) states by binding and unbinding to other receptors as they diffuse through the simulation space (Figure 3.2A). Off-rates for hetero- and homodimers are assigned probabilities based upon experimental measures for unoccupied and ligand bound dimers (Steinkamp et al., 2014). The catalytic activity of each monomer in a dimer is tracked throughout the simulation. Activity is dependent on the stochastically-governed orientation of the monomer in the asymmetric model, where one of the monomers is the “activator” and the other monomer is the “receiver.” Further, ErbB3 monomers are assumed to require phosphorylation by a “receiver” ErbB2 in a prior hetero-dimerization event. A phosphorylated ErbB3 monomer remains a competent “receiver” during subsequent encounters only until it is dephosphorylated. Simulation time steps are 1 × 10−6 s and observations are recorded every 0.05 s. Plots in Figure 3.2B show that dimerization is already occurring by the earliest observation interval and continues to rise over the first 10 s of the simulation. Phosphorylation kinetics are delayed, observable within 0.5 s of the simulation and rising to steady state values by 50 s.

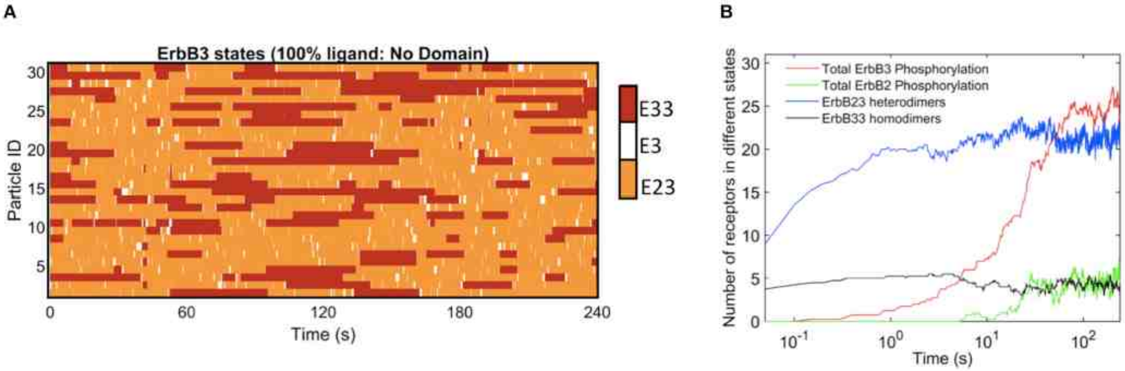


Figure 3.2: Kinetics of ErbB3 dimerization and phosphorylation. (A) Representative plot of individual ErbB3 receptors showing changes in receptor state over time. ErbB3 receptors cycle between homodimer, heterodimer and monomer states. (B) Plot showing the kinetics of dimer formation and phosphorylation of ErbB2 and ErbB3. ErbB2/3 heterodimer and ErbB3/3 homodimer formation are plotted with total ErbB2 and ErbB3phosphorylation over time for 100% ligand in the absence of domains. Data in B are the averages of 4 runs.

In Figure 3.3, we report the effect of adding domains to these simulations. The extreme cases of completely overlapping vs. non-overlapping ErbB2 and ErbB3 domains are shown in Figures 3.3A–H. Color keys in these plots indicate shifting profiles of monomers and dimers, as well as report phosphorylation states. Clearly, confinement in shared domains favors heterodimer interactions with a corresponding decrease in ErbB3 homodimers and ErbB2 monomers (Figures 3.3A,B). Phosphorylation kinetics is affected by co-confinement with a delayed but steep rise in phosphorylation (Figures 3.3C,D). Therefore, the overall signaling response is likely increased with shared domains.

Results in Figure 3.4 report dimers at steady state (240 s) using the three distinct domain configurations shown in Figures 3.1B–D as well as no domain configuration. Simulations with completely overlapping domains produced the greatest number of heterodimers regardless of ligand concentration, although the greatest difference can be seen with 100% ligand (Figure 3.4A). At lower ligand concentrations, the effect of overlapping domains on dimer formations was diminished. This phenomenon is best explained by segregation of the few ligand bound receptors. ErbB3 homodimers displayed the opposite trend to that of heterodimers, where the highest number of homodimers were seen when ErbB3 domains did not overlap with ErbB2 (Figure 3.4B). This was notable for conditions of 100% and 50% liganded ErbB3.

Steady state phosphorylation levels are also affected by the configuration of domains (Figures 3.4C,D). Phosphorylation levels of both ErbB2 and ErbB3 decreased as domain overlap decreased, highlighting the importance of heterointeractions for maximal signaling. ErbB2 phosphorylation was most affected by domain overlap, particularly in simulations with 100% liganded ErbB3 (Figure 3.4D). Note that ErbB3 phosphorylation, which is heavily dependent on interactions with ErbB2 is not favored under conditions where ErbB2 homodimers are predominant.

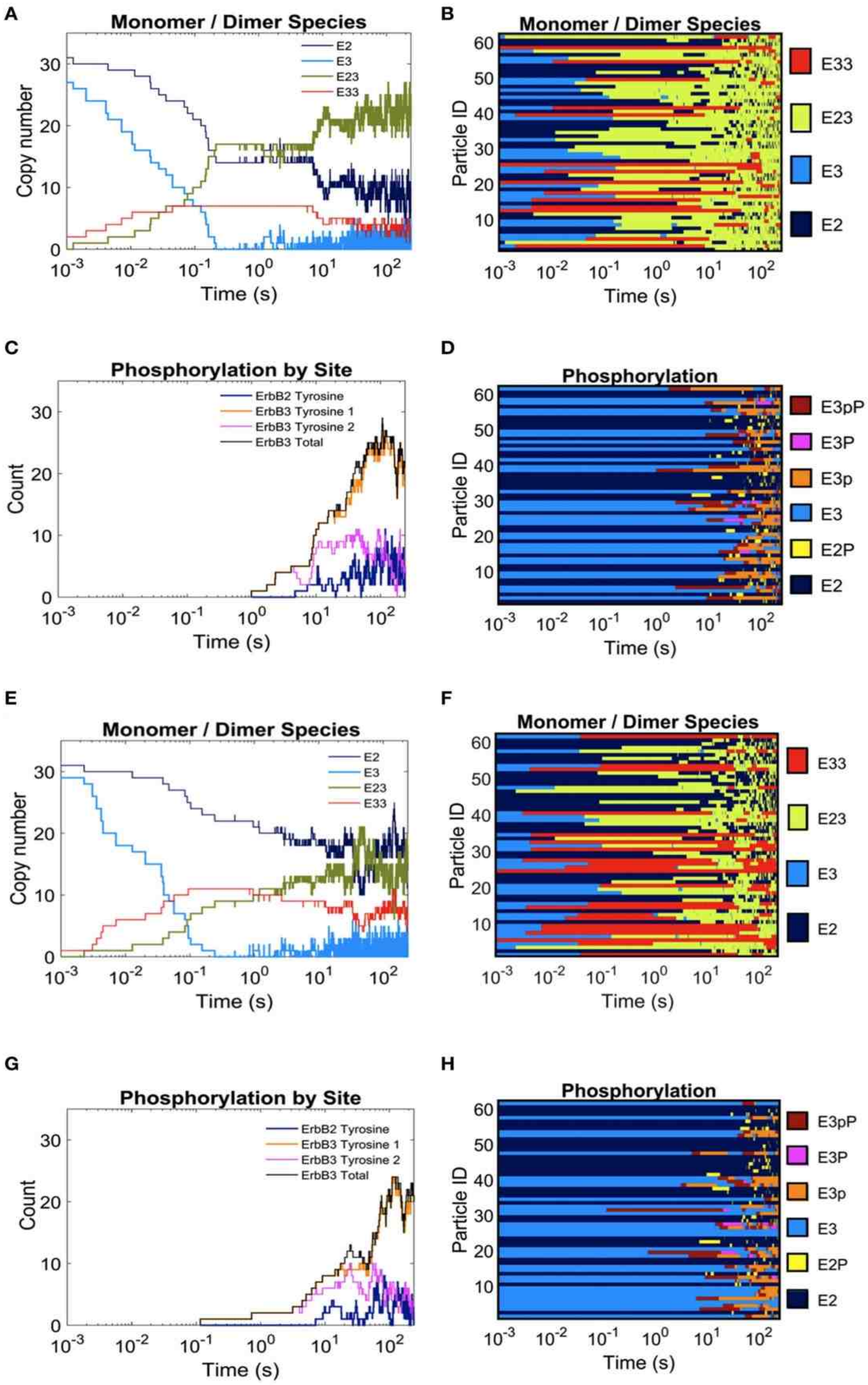


Figure 3.3: The effect of overlapping domains on ErbB2/ErbB3 dimerization and phosphorylation kinetics with 100% ligand-bound ErbB3. Plots for the completely overlapping domain configuration (A–D): The kinetics of dimer formation (A), representative plots of dimerization state for individual receptors over the simulation time (B), the kinetics of receptor phosphorylation (C), and a representative plot of phosphorylation state for receptors over time (D). (E–H): Plots for the non-overlapping domain configuration. Plots are arrayed as in (A–D).

ErbB2 and ErbB3 dimerization & phosphorylation, variable domain overlap

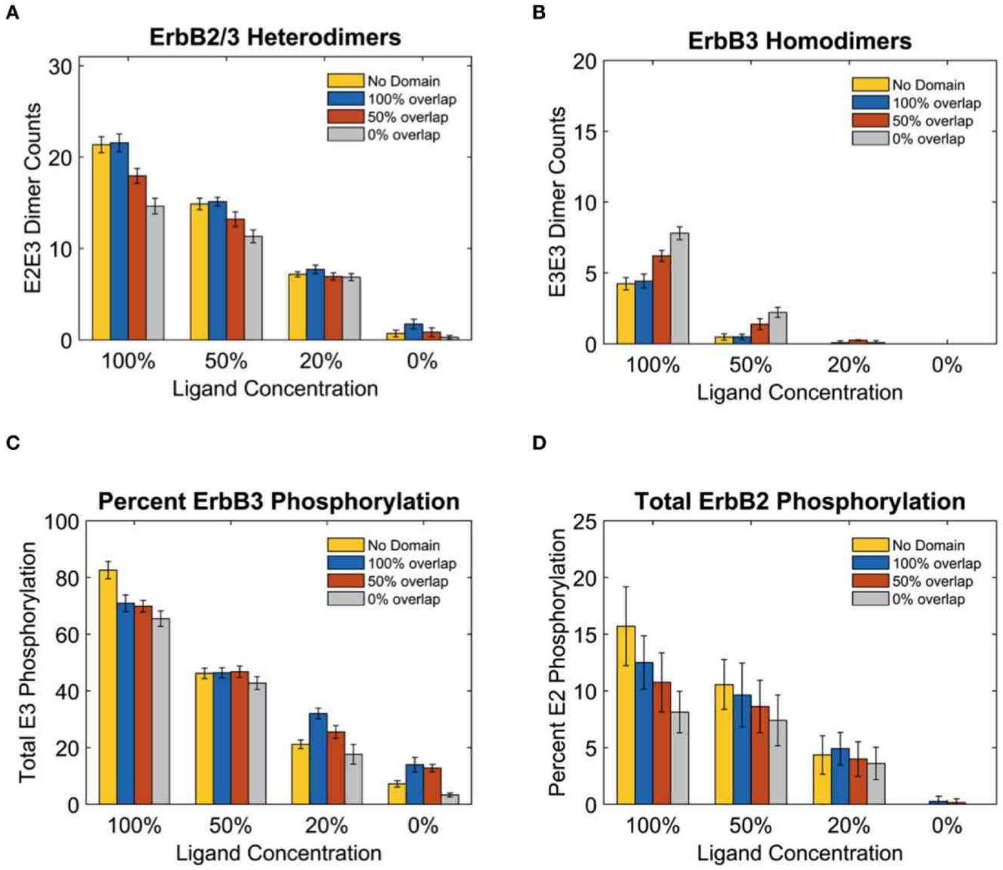


Figure 3.4: Overlapping domains influence dimer formation and phosphorylation. (A,B): Dimer counts across different ligand concentrations with 4 different membrane configurations- 100% (blue bars), 50% (orange bars), and 0% overlap (gray bars) as well as no domain simulations (yellow bars) for ErbB2/ErbB3 heterodimers (A) and ErbB3 homodimers (B). (C,D): Total receptor phosphorylation across different ligand concentrations and all four domain configurations for ErbB3 (C) And ErbB2 (D). All bars are the averages of 4 runs ± standard deviation.

3.6.2 Stronger domain retention affects receptor dimerization and phosphorylation events only when the domains partially overlap or non-overlap

Although the clustering of receptors in domains is important for signaling, little is known about the movement of receptors into and out of membrane domains or the extent to which this movement is altered with receptor activation. Since it is difficult to measure experimentally receptor residency times within domains, Pryor *et al.*, estimated an escape rate based on the ratio of domain-confined to free receptors in CHO cells under low ligand conditions (Pryor et al., 2015). To examine the effect of this parameter on signaling outcome, we ran simulations where we varied the escape rate to model changes in domain retention. The affinity of receptors for their domains was increased by reducing the escape rate of both monomers and dimers. We compared simulations run with the original nominal escape rate, or with the escape rate reduced by ½ or ¼. The effect of these escape rates were examined with different ligand concentrations in the four domain overlap configurations (Figure 3.5). Reducing the escape rates had no effect on heterodimer formation for domains that were completely overlapping. However, when the domains were partially overlapping or non-overlapping, heterodimer formation was significantly reduced as the escape rate decreased. For instance, in the case of 100% liganded ErbB3, when the escape rate was reduced to ¼ and the domains were partially overlapping, the number of heterodimers at steady state was 35% lower than with the original escape rate. With non-overlapping domains, heterodimers were reduced by 70% (Figure 3.5A). Similar trends were seen in 50% and 20% ligand conditions (Figures 3.5B,C). With unliganded ErbB3, heterodimerization was rare (Figure 3.5D). With completely overlapping domains, reducing the escape rates did not affect erbB3 homodimer formation either (Figures 3.5A–D). With overlapping domains, reducing the escape rate increased ErbB3 homodimers for partially and non-overlapping domains (Figures 3.5E–G). Escape rates ¼ of the original rate yielded maximum increase of 63%, which occurred with non-overlapping domains and 100% ligand (Figure 3.5E). Similar trends were seen with lower ligand concentrations (Figures 3.5F,G). Unliganded ErbB3 is not shown since there were no homodimers in this condition.

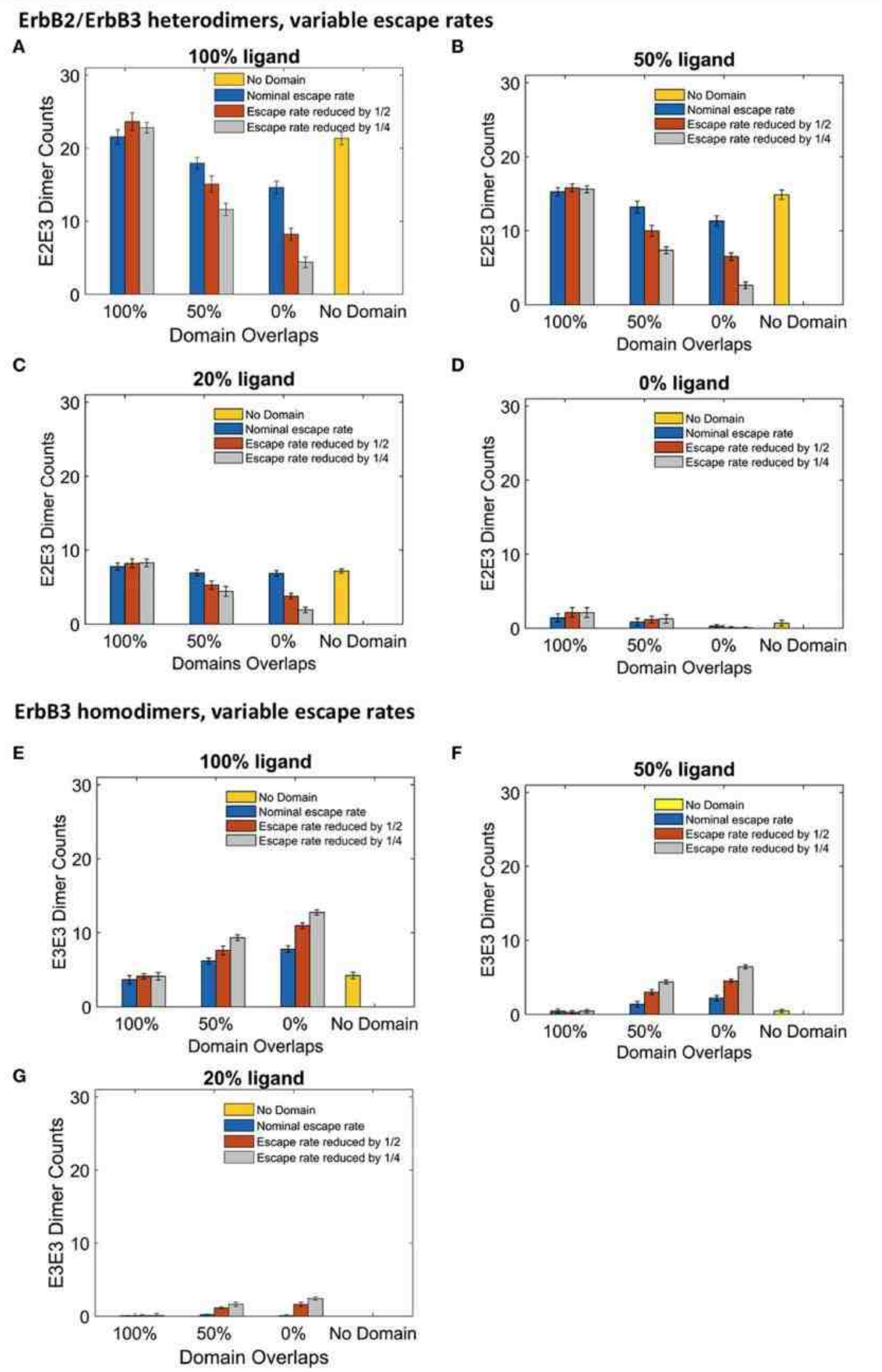


Figure 3.5: The effect of changes in domain retention on ErbB2/3 heterodimer and ErbB3/3 homodimer counts across different ligand concentration and domains. Dimer counts across different membrane configurations, ligand concentration and three different escape rates- nominal escape rate (blue bars), escape rate reduced by ½ (orange bars), and escape rate reduced by ¼ (gray bars) as well as no domain simulations (yellow bars). (A) ErbB2/3 heterodimer for 100% liganded ErbB3. (B)

ErbB2/3 heterodimer for 50% liganded ErbB3. (C) ErbB2/3 heterodimer for 20% liganded ErbB3. (D) ErbB2/3 heterodimer for 0% liganded ErbB3. (E) ErbB3/3 homodimer for 100% liganded ErbB3. (F) ErbB3/3 homodimer for 50% liganded ErbB3. (G) ErbB3/3 homodimer for 20% liganded ErbB3. The ErbB3/3 homodimer count was 0 for 0% liganded ErbB3. All bars are the averages of 4 runs ± standard deviation.

The significant changes in dimerization with increased domain retention had variable effects on downstream signaling as assessed by steady state phosphorylation levels of ErbB3 and ErbB2 (Figure 3.6). For ErbB3, phosphorylation levels are relatively stable with increased domain retention (Figures 3.6A–D). The greatest effect on phosphorylation levels occurred in the case of no domain overlap, where the ErbB3 monomers were more restricted from encounters with ErbB2. In the case of fully-liganded ErbB3, a four-fold reduction in escape rate led to a 28% reduction in phosphorylation (Figure 3.6A, gray bar for 0% overlap). For lower ligand concentrations, varying domain overlap had a greater effect on phosphorylation than domain retention (Figures 3.6C,D).

ErbB2 phosphorylation was markedly sensitive to increases in domain retention. Reduced ErbB2 phosphorylation corresponded to decreases in heterodimer formation (Figures 3.6E–G). Once again, little change was seen with completely overlapping domains. However, increasing domain retention lowered ErbB2 phosphorylation with either partially or non-overlapping domains. Results were striking for simulations run with a four-fold lower escape rate and 100% liganded ErbB3. Here, ErbB2 phosphorylation was reduced by 39% (partially overlapping domains) or 74% (non-overlapping domains).

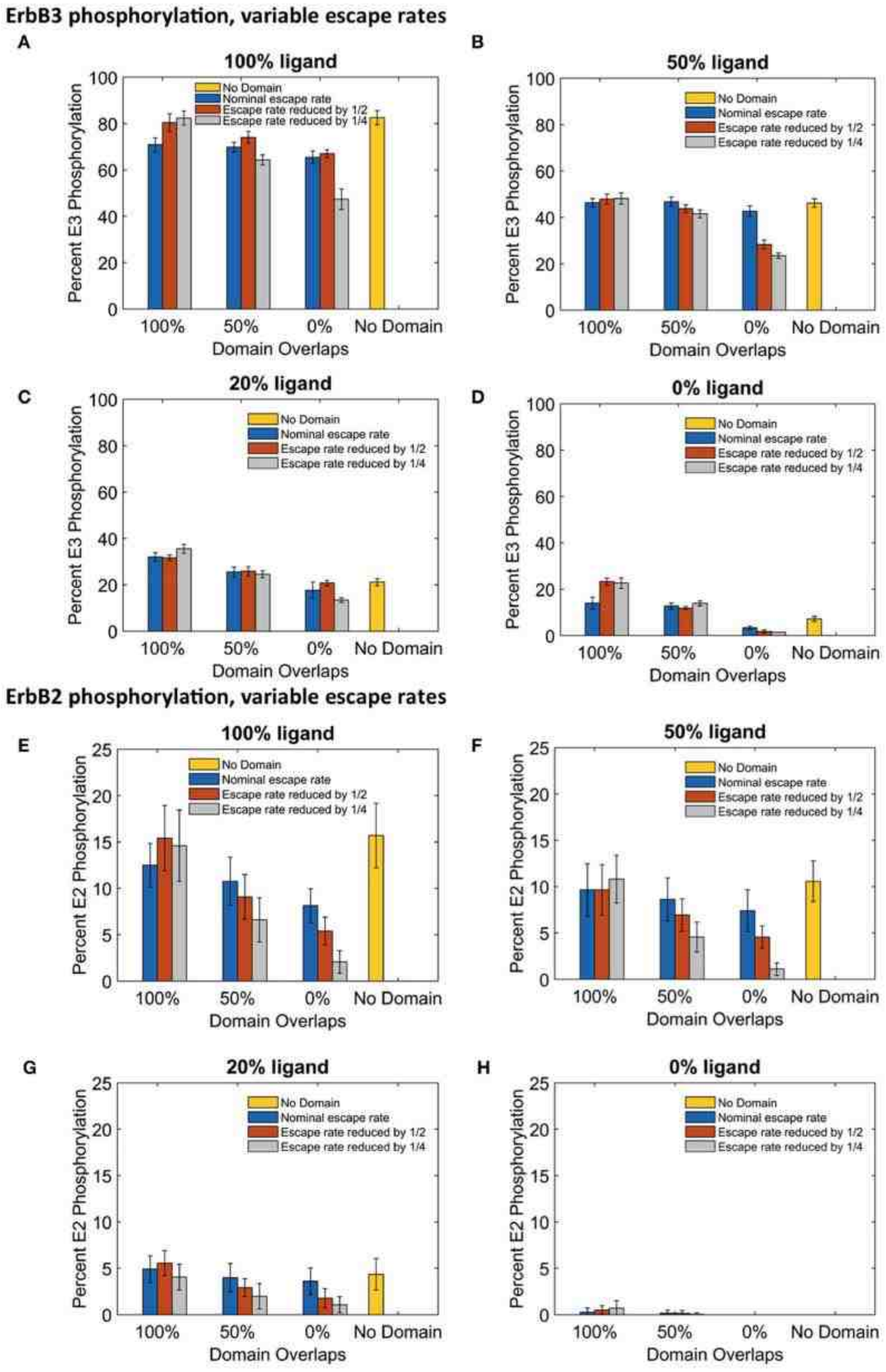


Figure 3.6: The effect of changes in domain retention on ErbB3 and ErbB2 phosphorylation across different ligand concentration and domains. Total receptor phosphorylation across different membrane configurations, ligand concentration and three different escape rates- nominal escape rate (blue bars), escape rate reduced by ½ (orange bars), and escape rate reduced by ½ (gray bars) as well

as no domain simulations (yellow bars). (A) Total ErbB3 phosphorylation for 100% liganded ErbB3. (B) Total ErbB3 phosphorylation for 50% liganded ErbB3. (C) Total ErbB3 phosphorylation for 20% liganded ErbB3. (D) Total ErbB3 phosphorylation for 0% liganded ErbB3. (E) Total ErbB2 phosphorylation for 100% liganded ErbB3. (F) Total ErbB2 phosphorylation for 50% liganded ErbB3. (G) Total ErbB2 phosphorylation for 20% liganded ErbB3. (H) Total ErbB2 phosphorylation for 0% liganded ErbB3. All bars are the averages of 4 runs ± standard deviation.

3.7 DISCUSSION

ErbB2 and ErbB3 are members of the ErbB family of receptor tyrosine kinases that are often co-expressed in cells. Under physiological conditions, neither receptor is active on its own. However, through heterointeractions these receptors activate two key pro-survival pathways. ErbB3 primarily activates the PI3K/Akt pathway and ErbB2 favors the MAP kinase pathway (Yarden and Sliwkowski, 2001). Activation of the ErbB2/ErbB3 signaling unit via overexpression of the receptors, gain-of-function oncogenic mutations, or autocrine release of the ErbB3 ligand, heregulin, have been identified in many types of cancer (Holbro et al., 2003; Wolf-Yadlin et al., 2006; Sheng et al., 2010; Jaiswal et al., 2013; Capparelli et al., 2015). Given the potency of this interaction, normal cells must maintain tight control over ErbB2/ErbB3 interactions. In the absence of ligand, dimerization is limited by the constant fluxing of the ErbB3 extracellular domain from a tethered, inactive conformation to an upright, active conformation with the active conformation stabilized by ligand binding (Dawson et al., 2007). Another way to control ErbB2/ErbB3 interactions may be through dynamic reorganization of membrane domains. Sequestration of ErbB2 and ErbB3 in separate domains could prevent spurious signaling in the absence of ligand, while reorganization into overlapping domains upon ligand binding could encourage the formation of signaling clusters (Vámosi et al., 2006). Evidence for reorganization can be seen in electron microscopy studies of SKBR3 breast cancer cell membranes. ErbB2 and ErbB3 are dispersed in the absence of ligand, but in the presence of ligand, ErbB3 forms large clusters with areas of co-localized ErbB2 and ErbB3 (Yang et al., 2007). It has also been shown that ErbB2 clusters within lipid rafts and that disruption of these rafts reduces both

ErbB2 clustering and the association of ErbB2 and ErbB3 (Nagy et al., 2002). The remodeling of domains during active signaling has not yet been explored by simulation, in part due to difficulties in accurately measuring the dynamics of these changes. Here, we have examined how domain remodeling, represented in our model by varying domain overlap and domain retention, will effect heterodimer formation and signaling.

Our spatial stochastic model of ErbB2/ErbB3 interactions provides a useful system in which to explore how changes in domain configuration might affect receptor activation. We began with a model parameterized based on single particle tracking data acquired under low (nanomolar) ligand conditions. We then explored how changes in domain characteristics, as well as ligand occupancy, influences dimerization and phosphorylation in this system. The sensitivity of the model to these parameters illustrates that variations in domain characteristics amongst different cell and tissue types are likely unappreciated modulators of signaling by these (and other) receptors.

Previous spatial stochastic models have shed insight on the effect of domains on signaling (Hsieh et al., 2008; Costa et al.,2009, 2011; Chaudhuri et al., 2011; Kalay et al., 2012). Kalay et al. evaluated movement of tracer molecules within lattice-based domains and found that confinement increased reaction rates (Kalay et al., 2012). Addressing ErbB receptor family interactions with rectangular subdomains, Hsieh *et al*., found that domains created local densities that favored EGFR interactions on the membrane surfaces (Hsieh et al., 2008). Our model increases the complexity by introducing two interacting receptor types with unique behaviors and overlapping, experimentally-defined domains. Thus, the model provides a mechanistic understanding of the interplay between domain overlaps and domain retention on the complex interactions of ErbB2 and ErbB3. The model relies on previously described characteristics of these receptors. For example, ErbB2 homodimers are not favored due to evidence for electrostatic repulsion (Garrett et al., 2003); this translates in the model to a low probability for ErbB2 homointeractions. In addition, ErbB3 has very low kinase activity unless activated by ErbB2 (Steinkamp et al., 2014). Thus, in cells where these are the two predominant ErbB species, they are predicted to be mutually dependent on each other for activation. It follows that differential preference of the two species for unique confinement zones or membrane domains should have a strong influence.

Accordingly, we found that phosphorylation of the two ErbB species was differentially affected by domain overlap. This was particularly evident in the case of 100% liganded ErbB3, where ErbB2 phosphorylation dropped by 50% between completely overlapping to non-overlapping domains (Figure 3.4D). At these physiological receptor levels, ErbB2 homo-encounters are largely unproductive due to the low on-rate. Simulations with more domain overlap had a larger number of heterodimer interactions than those with partial or no domain overlap. This was most notable when all ErbB3 were occupied with ligand (Figure 3.4). ErbB3 relies heavily on heterodimerization for activation. However, once ErbB3 receptors are activated by ErbB2, they can go on to homodimerize and activate other ErbB3 receptors. Therefore, steady state ErbB3 phosphorylation was less dependent on domain overlap.

It should be noted that the amount of hetero- and homodimers and phosphorylation levels were nearly the same between no domain spatial stochastic simulations and 100% domain overlapping conditions. This finding differs from our previous work with EGFR which showed that domains greatly improved phosphorylation of EGFR receptors, indicating that the introduction of multiple receptor types to these simulations further complicates outcome (Pryor et al., 2013). True domain overlaps are likely to fall somewhere between non-overlapping and completely overlapping configurations, indicating the need for spatial simulations that take this into account. Ligand binding to ErbB3 in SKBR3 breast cancer cell membranes leads to formation of large ErbB3 clusters with modest levels of co-localized ErbB2; this indicates that domain reorganization can occur during signaling (Yang et al., 2007). The remodeling of domains during active signaling has not yet been explored by simulation, in part due to difficulties in accurately measuring the dynamics of these changes.

SPT has revealed a range of non-brownian motion for proteins on the membrane plane. Anomalous diffusion is a term often used to explain the characteristic restricted movements of proteins that "hop" between membrane domains. There are also reports of specific membrane proteins that undergo directed (motor-driven) motion (Kusumi and Sako, 1996; Saxton and Jacobson, 1997; Schütz et al., 1997; Kusumi et al., 2005). These different modes of motion can have a profound impact on reaction kinetics on the membrane surface by perturbing reaction rates (Saxton and Jacobson, 1997; Melo and Martins, 2006). Thus, it is important to continue evaluating factors, such as diffusion coefficients, corral sizes and escape probability of proteins from their confined domains (Saxton and Jacobson, 1997), that are expected to impact signal initiation and propagation. In this work, we used a simulation approach to study the effect of escape probabilities on the reaction kinetics of the ErbB2/3 signaling pathway. We show that membrane segregation can influence signaling in non-intuitive ways that are linked to the individual characteristics of receptors. Given the technical challenges associated with measuring the dynamics of domain confinement, extent of mixing and escape rates in live cell membranes, simulation offers a powerful tool to explore these variables.

3.8 AUTHOR CONTRIBUTIONS

Conception and design: ÁH, JE. Development of computation framework: RK, ÁH. Acquisition and interpretation of data: RK, ÁH, MS, BW, JE. Writing, review, and/or revision of the manuscript: BW, RK, MS, ÁH, JE. Administrative support and study supervision: BW.

3.9 FUNDING

This study was supported by National Institutes of Health Grant P50GM085273 (BW), R01GM104973 (JE and ÁH), and R01HG006876 (JE).

3.10 ACKNOWLEDGMENTS

This work was funded by NIH P50 GM085273, awarded to the Spatiotemporal Modeling Center (Wilson). Use of the University of New Mexico Cancer Center Microscopy Facility and other shared resources supported by NIH P30CA118100 is gratefully acknowledged. We acknowledge the UNM Center for Advanced Research Computing (CARC) for access to high performance clusters.

CHAPTER 4: SPATIAL STOCHASTIC MODEL OF PRE-BCR

Romica Kerketta [1], Ádám M. Halász [2],Bridget S. Wilson [1,3] and Jeremy S. Edwards [3,4,5,6]

[1] Department of Pathology, University of New Mexico Health Sciences Center, Albuquerque, NM, USA, [2] Department of Mathematics and Mary Babb Randolph Cancer Center, West Virginia University, Morgantown, WV, USA, [3] Cancer Center, University of New Mexico Health Sciences Center, Albuquerque, NM, USA, [4] Department of Chemical and Biological Engineering, University of New Mexico, Albuquerque, NM, USA, [5] Department of Chemistry and Chemical Biology, University of New Mexico, Albuquerque, NM, USA, [6] Department of Molecular Genetics and Microbiology, University of New Mexico, Albuquerque, NM, USA.

4.1 ABSTRACT

Progenitors of B cells express the pre-B cell receptor (pre-BCR) early in the B lymphocyte development pathway. Expression of this receptor is critical for survival and proliferation of B cells. The pre-BCR undergoes ligand independent tonic signaling through frequent, but short lived, homodimer interactions. To investigate tonic signaling emanating from this receptor, we developed a rule-based spatial stochastic model of pre-BCR aggregation and downstream signaling events. The model was populated with data from single particle methods from two different pre-BCR cell lines (697 and Nalm6), which exhibit characteristic differences in their diffusion coefficients and dimer off rates. We found that these differences affected pre-BCR aggregation and consequent signaling events in the pre-B cells. The Nalm6 cell line, which had a lower off rate and lower

diffusion coefficient, formed higher order oligomers than the 697 cell line. There was also an increase in the steady-state levels of receptor phosphorylation in the Nalm6 cell line. Thus, the spatial stochastic model of pre-BCR presented here was able to estimate aggregate sizes and predict the receptor phosphorylation landscape during tonic signaling.

4.2 INTRODUCTION

The precursor B cell receptor (pre-BCR) appears early in the developmental pathway of B lymphocytes and serves as a checkpoint for the progression of B cell progenitors into mature B lymphocytes (Rickert, 2013) . The pre-BCR is expressed on the surface of progenitor B cells and it is composed of a rearranged Immunoglobulin heavy chain (IgH) and non-polymorphic surrogate light chain (SLC) consisting of λ5 and VpreB (Rickert, 2013). The pre-BCR is also non-covalently attached to Igα (CD79a) and Igβ (CD79b), two heterodimeric subunits containing the immunoreceptor tyrosine-based activation motif (ITAM) that help to propagate signaling downstream of the pre-BCR (Monroe, 2006). Signaling from the pre-BCR entails phosphorylation of the ITAMs on tyrosine residues by Src family kinases (SFK) such as Lyn; the phosphotyrosines then serve as docking sites for the spleen tyrosine kinase (Syk) (Gauld and Cambier, 2004). Syk docks to the pre-BCR using its pair of Src homology 2 (SH2) domains and generates signaling responses that lead to remodeling of the cytoskeleton, intracellular calcium response and differential gene expression patterns necessary for B cell maturation (Cornall et al., 2000; Guo et al., 2000; Monroe, 2006).

The SLCs of the pre-BCR do not undergo gene rearrangements (Monroe, 2006). Since the SLC are non-polymorphic, the pre-BCR lack the ability to bind to conventional antigens of the BCR, hence, one of the major challenges in the field has been to identify the exact mechanism with which the pre-BCRs interact and activate signaling cascades that promote progenitor B cell differentiation and survival. There is limited evidence for ligand dependent pre-BCR signaling, that is mediated through reactivity to self-antigens; crosslinking can also be induced when λ5 components of the SLC bind to the dimeric stromal ligand galectin-1 (Gauthier et al., 2002; Kohler et al., 2008; Erasmus et al., 2016). Basal signals may emanate from Igα and Igβ on unaggregated receptors (Fuentes-Panana et al., 2004). Ubelhart and colleagues propose that pre-BCR signaling is dependent on a conserved asparagine (N)-linked glycosylation site on IgH (Ubelhart et al., 2010). However, there is growing evidence that ligand-independent pre-BCR homo-interactions lead to induction of weak or 'tonic' signaling on progenitor B cells (Ohnishi and Melchers, 2003; Monroe, 2006; Bankovich et al., 2007; Erasmus et al., 2016b). Erasmus *et al*., used single particle tracking (SPT) methods to track the diffusional dynamics as we all as homodimer events between ligand independent pre-BCR on B-cell precursor acute lymphoblastic leukemia (BCP-ALL) cell lines. The pre-BCR on these cell lines undergo frequent but short lived dimerization events which lead to tonic signaling comprising of induction of pro-survival B cell lymphoma 6 protein (BCL6), a transcription repressor, necessary for pre-BCR to transition into the next developmental stage (Duy et al., 2010; Erasmus et al., 2016).

Acute lymphoblastic leukemia (ALL) constitutes one of the most common childhood cancers and a majority of these neoplasms have been found to lack a functional pre-BCR (~85%) (Müschen, 2015). However, there is a subset of ALL cases (13.5%), where the neoplasms express a functional pre-BCR and exploit the tonic signaling generated foirom this receptor to survive and proliferate (Geng et al., 2015; Müschen, 2015). Inhibitors of Lyn and Syk tyrosine kinases, which are active participants in the pre-BCR pathway, have been shown to negatively affect survival of pre-BCR+ ALL cells, thus highlighting the therapeutic potential of these small molecules in combating BCP-ALL (Geng et al., 2015; Erasmus et al., 2016). Erasmus et al also observed that monovalent anti-VrepB antibody fragments could inhibit pre-BCR dimerization and interrupt survival signals from this receptor (Erasmus et al., 2016).

Key parameters in our model are derived from Erasmus *et al.*, This includes distinct values for pre-BCR diffusion coefficients, as well as homodimer off rates, for two BCP-ALL cell lines: 697 and Nalm6 (Erasmus et al., 2016). The pre-BCR on the surface of 697 cells dimers diffused considerably faster and had higher dimer off rates as compared to the Nalm6 cells. These data led us to hypothesize the difference in dimer off rate could lead to existence of higher order oligomers in Nalm6 cell lines; the tendency to form slightly larger aggregates would explain the overall slower diffusion rate. In support of this theory, some of the apparent dimer pairs observed through SPT of pre-BCR in the Nalm6 cell line were also more than 100nm apart, which was the theoretical distance between a single pair of dimerized receptors labeled with quantum dots. One interpretation of these data is that the SPT captured cases of Nalm6 oligomers where the two quantum dots were bound to receptors located at the ends of a chain of a trimer or tetramer.

Since receptors are sparsely labeled in SPT experiments, aggregation beyond dimers is difficult to quantify. To investigate the existence of these higher order oligomers and the impact of the apparent diffusion coefficients and dimer off rates, we developed a spatial, stochastic model of pre-BCR aggregation and tonic signaling. We parameterized the model with coefficients directly measured from single particle tracking as well as from the available literature. Our motivation for creating a spatial model of tonic pre-BCR signaling pathway evolved from observing electron micrographs of labeled pre-BCR which were confined in domains formed by the actin cytoskeletons (Figure A.1). Using the spatial stochastic model, we found that receptor dimer off rates and domains affected aggregate sizes and consequent signaling events.

4.3 MATERIALS AND METHODS

We developed a rule-based spatial stochastic model of pre-BCR to investigate tonic signaling occurring downstream of receptor aggregation events. The two pre-BCR cell lines that were used for experimental measurements were Nalm6 and 697, each with specific pre-BCR diffusion coefficients as well as dimer off rates (Table 3). This model has been parameterized with data from SPT measurements and morphometric analysis of pre-B cell lines, as well as from the literature. The earliest tyrosine kinases in the signal transduction pathway, Lyn and Syk, are explicitly represented in the model. Details of reaction kinetics and rules specifying the interaction of molecules are given below.

4.3.1 The signal transduction pathway

Although the pre-BCR and BCR occur at distinct time points in the lymphocyte development and contain structurally different light chains, both the receptors enlist the same set of Src and Syk family kinases to initiate signaling (Benschop and Cambier, 1999; Meffre et al., 2000). The Pre-BCR propagates signaling through the non-covalently attached Igα and Igβ units which each contain ITAMs (Monroe, 2006). The ITAMs contain tyrosine residues (Y182 and Y193 on Igα and Y195 and Y206 on Igβ) that are substrates for phosphorylation by the kinases (Pao et al., 1998; Storch et al., 2007).

In the model, the pre-BCR receptors form linear dimers which may bind further to form higher order oligomers through their free receptor binding domains (Table 3). Receptor aggregation are triggered through molecule collision events and receptor unbinding is intrinsically triggered whose probability is determined by the experimentally measured dimer off-rate and the time step.

Table 3: Pre-B cell receptor cell line characteristics in the model

	697	Nalm 6
No of receptors[a]	10000	
No of Lyn available to receptor[b]	10% of receptor	
No of Syk[a]	48,290	274,302
Binding radius (μm)[c]	0.000215	
Dimer off rate (/s)[c]	0.772	0.164
Diffusion coefficient Receptors ($\mu m^2/s$)[d]	0.16	
Diffusion coefficient Lyn ($\mu m^2/s$)[e]	0.4	
Diffusion coefficient Syk ($\mu m^2/s$)[f]	17	

[a] Experimental data in this paper

[b] Estimated from Wofsy et al. 1997, Faeder et al. 2003

[c] Estimated from Erasmus et al. 2016

[d] Erasmus et al. 2016

[e] Stone et al. 2015

[f] Brock et al. 1999

Ligand independent receptor aggregation in pre-BCR is followed by binding of Lyn to the ITAMs (Monroe, 2006; Erasmus et al., 2016). Lyn has four structurally distinct domains through which it interacts with the ITAMs (Boggon and Eck, 2004; Parsons and Parsons, 2004). The unique domain of Lyn is known to constitutively associate with receptors and bind to non-phosphorylated Igα whereas the SH2 domain binds to phosphorylated ITAMs on both Igα and Igβ (Pleiman et al., 1994; Vonakis et al., 1997). Although Lyn has been reported to bind ITAMs in both signaling subunits, it preferentially binds to Igα at least twice more likely than to Igβ (Johnson et al., 1995).

In the model, Lyn can bind to a free ITAM site on a receptor (Table 4). The receptor maybe a part of an aggregate or be a single receptor by itself. Lyn can bind to an unphosphorylated Igα through its unique domain and bind to a phosphorylated Igα or phosphorylated Igβ through its SH2 domain. Only one Lyn molecule is allowed to bind per Igα or Igβ. Just as receptor-receptor binding events occur, Lyn-receptor binding events are collision triggered and unbinding is based on a probability calculated through off rates and the time step. The two phosphorylation sites on Igα have been lumped together into one site and can have the phosphorylation status- 0, 1 or 2. The phosphorylation sites on Igβ are treated similarly.

Table 4: Lyn and Syk molecule binding radii and dimer off rate

Molecules	ITAM	Phos. Status	Binding radius (μm)	Dimer off rate (/s)
Lyn	Igα	0	2.29E-04[a]	20[b]
		≥ 1	2.29E-04[a]	0.12[b]
	Igβ	≥ 1	1.14E-04[c]	0.12[b]
Syk	Igα	1	1.31E-04[d]	2.6[e]
		2	1.57E-03[d]	0.3[e]
	Igβ	1	4.37E-05[f]	2.6[e]
		2	5.25E-04[f]	0.3[e]

[a] Estimated from Faeder et al. (2003) and Smoldyn- Andrews et al. (2004)

[b] Faeder et al. (2003)

[c] Estimated according to observed experimental data (Lyn binding to Igβ is approximately 1/2 of the binding observed to Igα) in Johnson et al. (1995).

[d] Estimated from Schwartz et al. (2017), Tsang et al. (2008) and Smoldyn- Andrews et al. (2004)

[e] Schwartz et el. (2017)

[f] Estimated from Schwartz et al. (2017), Tsang et al. (2008), Kurosaki et al. 1995 (Binding of Syk to Igα is 3x more than binding observed to Igβ) and Smoldyn- Andrews et al. (2004)

Lyn association with the receptors is followed by receptor phosphorylation and one Lyn in an aggregate is sufficient for initiating phosphorylation of other receptors in an aggregate (Wofsy et al., 1999). Lyn itself can also become transphosphorylated

(Y397) by other Lyn molecules, which results in an increase of Lyn kinase activity (Yamashita et al., 1994; Sotirellis et al., 1995; Wofsy et al., 1999; Ingley, 2012).

In the model, Lyn on a receptor can phosphorylate Lyn on a nearby receptor, i.e. the two receptors must be bound to immediate neighbors in the same receptor complex (Table 5). Phosphorylation rates depend on the phosphorylation state of Lyn. Similar to Lyn-Lyn phosphorylation, Lyn must be on an immediately adjacent receptor in the same complex in order to phosphorylate the ITAMs. Phosphorylated Lyns are assumed to be activated and have a stronger kinase activity. The Lyn phosphorylation site can have the phosphorylation status- 0 or 1.

Table 5: Kinase and substrate phosphorylation status and rates

Kinase	Phos. Status	Substrate	Phos. Status	Rate (/s)
Lyn	0	Igα(Y182/Y193)	0	30[a]
	0		1	15[b]
	1	Igβ(Y195/206)	0	100[a]
	1		1	50[b]
	0	Lyn(Y397)	0	30[a]
	1			100[a]
	0	Syk(Y342/Y346)	0	30[a]
	1			100[a]
Syk	0	Syk(Y519/Y520)	0	100[a]
	1			200[a]

[a] Faeder et al. (2003)

[b] Estimated from Barua et al. (2012). Phosphorylation of 2nd site on the ITAM occurs at half the rate from first.

The phosphorylated ITAMs form docking sites for the Syk (Kurosaki et al., 1995). While Syk can bind to both mono-phosphorylated and doubly-phosphorylated ITAMS through its SH2 domains, the binding affinity is significantly higher for doubly phosphorylated ITAMS; this is reflected by high dimer on-rate and lower off-rate (Tsang et al., 2008). Binding of Syk to Igα is three times more than the binding seen to Igβ (Kurosaki et al., 1995).

In the model, free Syk can bind to any one of the two ITAM (Igα and Igβ) sites on each receptor upon collision (Table 4). Each ITAM in an ITAM pair is regarded as a Syk binding site, and can bind Syk independently of each other. However, since Lyn binding is also possible on the same sites, Syk molecules cannot bind the ITAMs if they are occupied and the same rule is applied for Lyn molecules. Syk dissociation from the receptor also occurs through an intrinsically triggered unbinding probability which is determined through the Syk off rate and the time step. The Syk off-rate is higher for doubly phosphorylated ITAMS versus singly phosphorylated.

Syk can undergo phosphorylation on specific tyrosine residues in its catalytic domain (Y519 and Y520) by another Syk docked on adjacent ITAMS in the same BCR or pre-BCR aggregate (Keshvara et al., 1998; Zhang et al., 2000). Syk can also undergo phosphorylation by a Lyn docked on adjacent ITAMs in the same BCR or pre-BCR aggregate molecule in its linker regions (Y342 and Y346) (Keshvara et al., 1998).

In the model, Syk or Lyn molecules must be on an immediately adjacent receptor in the same complex in order to phosphorylate other Syk molecules (Table 5). Syk molecules can phosphorylate other Syk molecules in their catalytic domain and Lyn molecules can phosphorylate other Syk molecules in their linker regions. Syk molecules that have been phosphorylated in their catalytic domain are assumed to have stronger kinase activity. The two phosphorylation sites on the catalytic domain have been lumped together and can have the phosphorylation status- 0 or 1. The two phosphorylation sites on the linker region have been treated similarly

Receptor, Lyn and Syk dephosphorylations in the model are all intrinsically triggered from a dephosphorylating probability which takes into account the molecule dephosphorylations rate and the time step (Table 6).

Table 6: Dephosphorylation rates

Substrate	Phos. Status	Rate (/s)
Igα(Y182/Y193)	2	40[a]
	1	20[b]
Igβ(Y195/206)	2	40[a]
	1	20[b]
Lyn(Y397)	1	20[b]
Syk(Y342/Y346)		
Syk(Y519/Y520)		

[a]Estimated from Faeder et al. (2003) and Barua et al. (2012)(Dephosphorylation of doubly phosphorylated ITAMs occur at 2x the rate of singly phosphorylated ITAM)
[b]Faeder et al. (2003)

4.3.2 Simulation Landscape

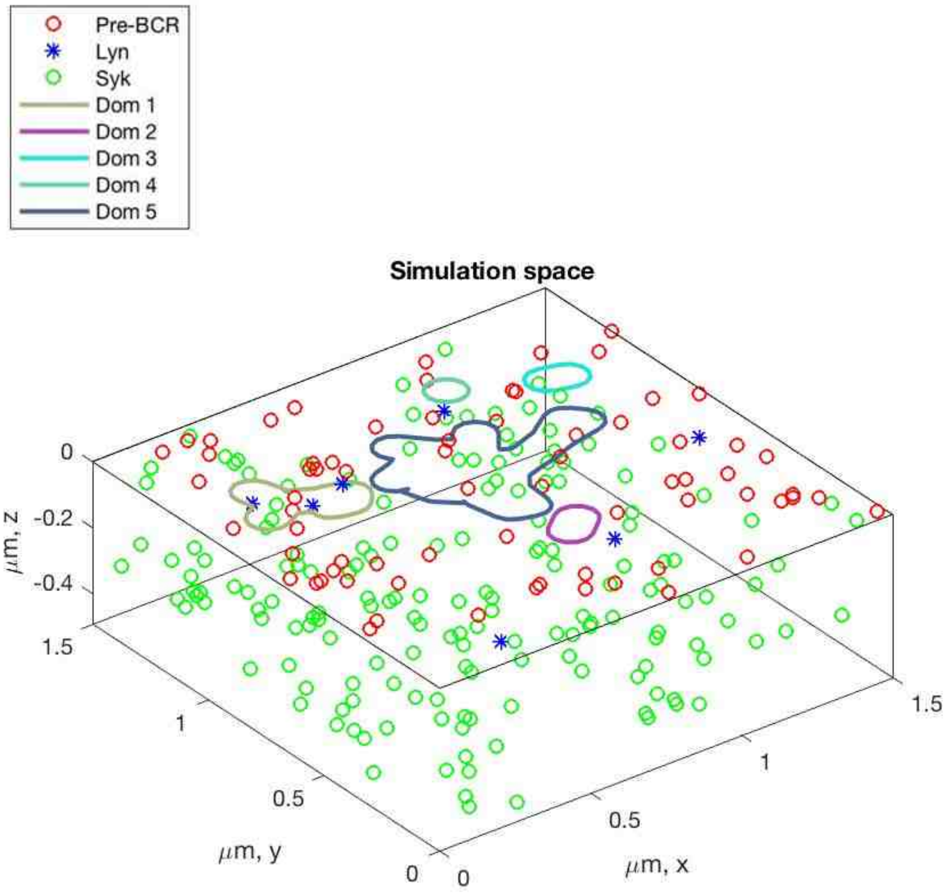


Figure 4.1: 3-D simulation space containing pre-BCR, pre-BCR domains, Lyn and Syk molecules

We created a 3-D simulation landscape in MATLAB to track interactions between the pre-BCR and Lyn, which are both plasma membrane residents, and Syk in the cytosol. Thus, the 2-D membrane is populated with transmembrane pre-BCR and inner leaflet-bound Lyn molecules, while the 3-D cytosol is populated with Syk molecules (Figure 4.1). The membrane contains pre-BCR receptor specific domains and receptors can diffuse across domains and domain-free areas. Figure 4.1 depicts domains that were identified through analysis of two-color SPT. For SPT measurements, monovalent Anti-Igβ Fabs' against the Igβ subunit were generated using hybridomas and labeled with streptavidin-conjugated QD585 and QD655. The monovalent QD probes were then allowed to tag Pre-BCR and their movement was tracked on a live membrane using SPT imaging (Erasmus et al., 2016). In order to extract the receptor domain sizes and contours from SPT, data sets containing the particle trajectories were subjected to the domain reconstruction algorithm (DRA), which was previously developed and used in Pryor *et al*., 2015 (Pryor et al., 2015). Briefly, the DRA reads in the SPT trajectories and ranks them into slow moving (confined) or fast moving points (free) using their jump sizes over different time intervals. The confined points are postulated to be in a domain (such as lipid rafts, protein domains or corrals) that impede the movement of the particles. We then cluster the slow moving points into groups based on whether their distance from each other is less than the reference distance, L. After cluster identification, we build contours around them that represent vertices of receptor domains.

From morphometric measurements, pre-B cell was estimated to have a total cell surface area of 315.7 μm^2 and a cytosolic volume of 321.8 μm^3. The pre-B cell radius was 5 μm and nuclear radius was 3.7 μm. In our simulations, the pre-B cell membrane landscape is represented by a rectangular cuboid with an area of 2.25 μm^2 (1.5 μm x 1.5 μm) and volume of 1.25 μm^3 (depth of 0.5 μm). The total number of receptors is equivalent in both the cell lines (10,000 receptors) for a density of ~ 32 receptors/μm^2. For our simulation area, this resulted in 71 receptors occupying the membrane landscape.

The amount of Syk varied between the cell lines. Based upon calibrated western blotting experiments, we estimate that 697 cell line has ~ 48,000 Syk molecules while the Nalm6 cell line has 274,000. For our simulation volume, this resulted in 169 Syk molecules for the 697 cell line and 959 Syk molecules for the Nalm6 cell line. The amount of Lyn available to the receptors is estimated to vary between 5% and 10% of the total receptors present in the cell (Yamashita et al., 1994; Wofsy et al., 1997). Faeder *et al.*, in their investigation of signals emanating from FcεRI, modeled the pool of Lyn available to the receptors as 7% of the total receptor concentration and assumed that all of the available Lyn is in a form, which when bound to the receptors, is capable of initiating phosphorylation events (Faeder et al., 2003). We apply the same principles in our model, where we assume that the total amount of Lyn available to the receptors would be at most 10%. This would mean that at any given time in the pre-B cell, only 1000 Lyn molecules are available for interaction with the 10,000 pre-BCRs. For our simulation area, this amounts to 7 Lyn molecules available for interaction with a total number of 71 receptors.

4.3.3 Molecule diffusion

In our simulation molecules can undergo Brownian motion in the x, y and z plane. Lyn and pre-BCR undergo diffusion in the x and y plane only, whereas Syk molecules can also diffuse in the z plane.

Receptor jumps are generated by choosing a random number from a normal distribution and processing it with the root mean square (RMS) of the molecule to generate the new coordinates with the time increment of Δt. The RMS is given by: $RMS = \sqrt{2 * Diffusion\ Coefficent * \Delta t}$. Following rules are applied for calculation of the new spatial coordinates (Andrews and Bray, 2004; Erban, 2014; Pryor et al., 2015):

$$x(t+\Delta t) = x(t) + RMS * \xi x$$

$$y(t+\Delta t) = y(t) + RMS * \xi y$$

$$z(t+\Delta t) = z(t) + RMS * \xi z$$

ξx, ξy, ξz are random numbers chosen from a normal distribution. x, y and z represent the molecule’s Cartesian coordinates.

Since pre-BCRs form higher order oligomers, in the model we assume that the diffusion of a pre-BCR complex is inversely proportional to the size of the complex. The size of the complex reflects the number of receptors in an aggregate.

4.3.4 Boundary conditions and probability of escape

For any simulation space, boundary conditions need to be specified so that particles remain in the simulation area or volume. For the pre-BCR, we apply periodic boundary conditions at the edges of the simulation space. When a receptor reaches the edge of the simulation space (in x or y plane) and the receptor jump calculated displaces the receptor outside the simulation space, we divide the jump into two segments. The first segment displaces the receptor to the edge and the second segment is calculated from the opposite edge of the simulation space, such that the receptor re-enters the simulation space from the opposite boundary. For receptors in domain we apply the reflective boundary conditions. As before, the receptor jump is divided into two segments: one segment displaces the receptor to the edge of the domain and the second displaces it outside of the domain. When the receptor reaches the edge of the domain, a probability for escape from the domain is calculated and if the probability of escape is not met, then the receptor is reflected back into the domain with the remaining segment. If the probability of escape is met, then the receptor continues across the domain. An exit penalty limits receptor escape from the domains. The exit penalty was obtained by calculating the ratio of the membrane area explored by the slow moving points versus the membrane area explored by fast moving points. The exit probability for receptors in the pre-B cells was found to be 0.2. We apply the same periodic boundary conditions to Lyn and Syk molecules in the x and y plane. For Syk, we also apply reflective boundary condition in its z plane, such that when the receptor reaches edge of the simulation space, it is reflected back into the cytosol.

4.3.5 Reaction kinetics

For molecule reactions, we chose reaction kinetics similar to those used in the spatial stochastic simulator Smoldyn, which uses Smoluchowski dynamics with revisions to implement reaction events. In essence, there are two different ways to simulate reaction events depending on whether they are first or second order reactions. First order reactions include molecule phosphorylation, dephosphorylation and molecule dissociation. The probability of any of these first order reaction events occurring at time step Δt is given by (Andrews and Bray, 2004; Pryor et al., 2015):

$$P(_{\text{First order reaction}}) = 1 - \exp\left(-\text{First order reaction rate} * \Delta t\right)$$

However, due to the very small time step, this probability has been reduced to the following:

$$P(_{\text{First order reaction}}) = \text{First order reaction rate} * \Delta t$$

Second order reactions include molecule association events such as receptor dimerization or aggregation into higher order oligomers and molecules such such as Lyn and Syk binding to the receptor. For these reactions, a parameter called the binding radius is used to determine the outcome of a collision event. If two molecules are within the binding radius of each other, then a binding event will take place. The binding radius for Lyn and Syk molecules were calculated using Smoldyn, which takes into account the on rate of the reaction, diffusion coefficients and the time step to determine a binding radius for molecular association events. The binding radius for a pair of receptors was determined using SPT data from Erasmus *et al*., well mixed Matlab simulations and spatial stochastic simulations with no domains. Briefly, we ran well mixed Matlab simulations with an estimated on rate and estimated dimer off rate for pre-BCR aggregation in the 697 cell line. This produced a ratio of monomers, dimers and higher order oligomers. We then used the estimated dimer off rate with varying binding radii in a spatial stochastic simulation (coded in Fortran) with no domains to obtain the same ratio of monomers, dimers and higher order oligomers as observed in the well mixed Matlab simulations. The binding radius which correctly reproduced the ratio of the oligomers in the spatial simulations was used in the simulations as the pre-BCR binding radius (Table 3).

4.4 RESULTS

4.4.1 Impact of varying dimer off rate and domains on receptor aggregation

From SPT measurements, Erasmus *et al.*, had observed that the 697 cell line had a higher dimer off rate as compared to the Nalm6 cell line. This led to the speculation that Nalm6 was forming higher sized oligomers, supported by observed cases of SPT that captured apparent "dimers" of pre-BCR tagged quantum dots that exceeded the theoretical distance of 100 nm for a minimal pre-BCR dimer. We populated our spatial stochastic model with the estimated off rates in order to observe whether the two cell lines formed different sized aggregates. Figure 4.2A represents the aggregate sizes formed in the 697 cell line and Figure 4.2C represents the aggregate sizes formed in the Nalm 6 cell line. As observed from the Figures 4.2A and C, the Nalm6 cell line forms much higher order oligomers when compared to the 697 cell line. The number of monomers was also higher in the 697 cell line as compared to the Nalm6. Thus, the spatial stochastic models provides evidence for the existence of much higher order oligomers in Nalm6 cell line as compared to 697.

From electron microscopy images of immune-gold labeled plasma membrane "rip-flips" (Figure A.1), we observed corral-like cortical cytoskeletal structures on the membrane of pre-B cells indicating that the receptors might be confined in domains. In order to investigate the effect of domains on receptor aggregation, we used data gathered from SPT measurements to re-create pre-BCR domains in our *in silico* membrane landscape. We found that the presence of domains increased the size of pre-BCR aggregates in both the cell lines (Figure 4.2A-D). Figure 4.2A displays the ratio of aggregates in the 697 cell line with domains while Figure 4.2B displays the ratio of aggregates formed in the same cell line without domains. Clearly there is an increase in the oligomer size in simulations where domains were present. Figure 4.2C and Figure 4.2D compare the aggregate sizes in the Nalm6 cell line, with and without domains respectively. A similar trend is seen where the simulations with domains had an impact on the aggregate sizes. Hence, as seen in our earlier studies (Kerketta et al., 2016), domains again prove to be important regulators of receptor aggregation and consequent signaling pathways.

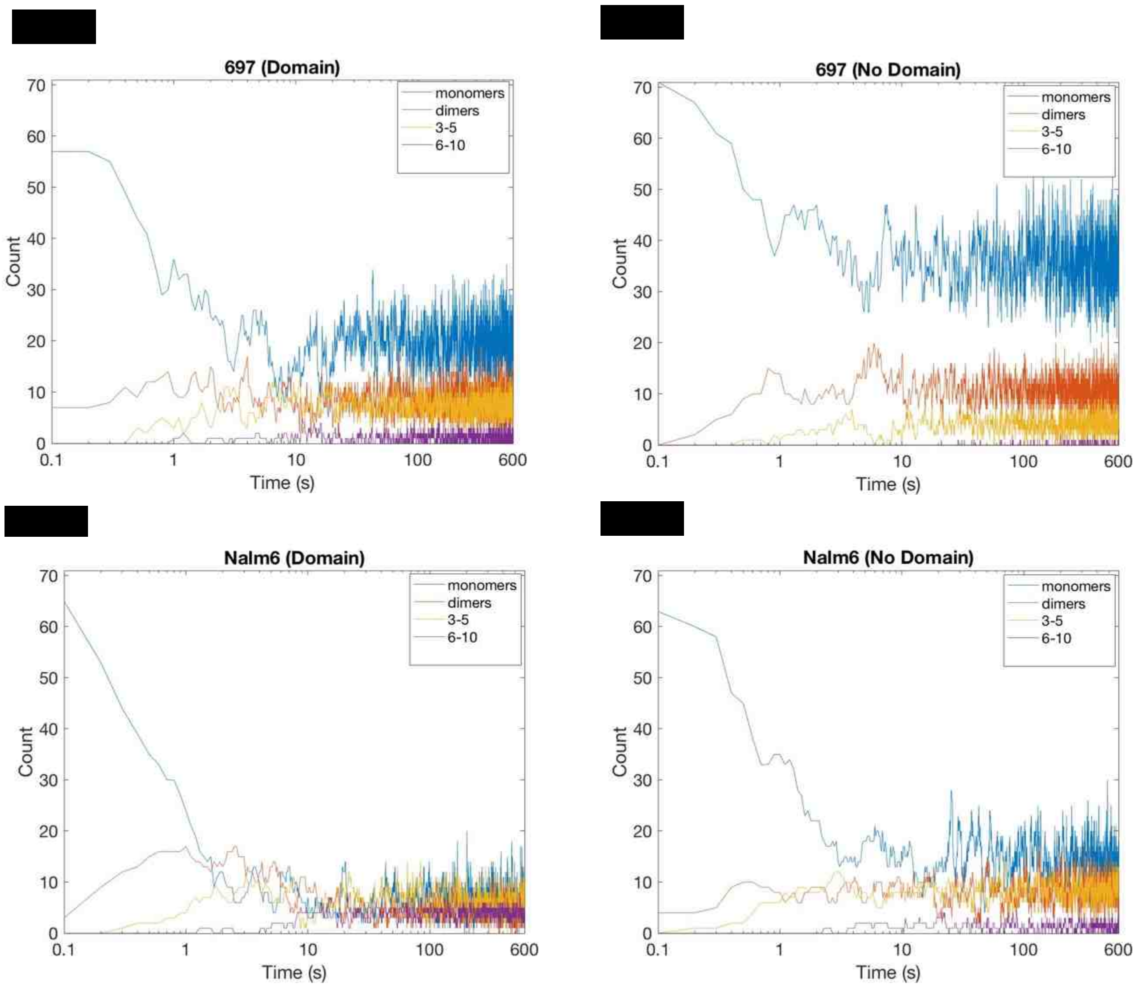


Figure 4.2: Receptor aggregation on 697 and Nalm6 cell line. (A) 697, Domain. (B) 697, No Domain. (C) Nalm6, Domain. (D) Nalm6, No Domain.

4.4.2 Impact of varying dimer off rate and domains on ITAM phosphorylation

It is known that receptor aggregation leads to phosphorylation of tyrosine residues on the ITAMs of pre-BCR by Lyn as well as Lyn transphosphorylation by other Lyn molecules (Wofsy et al., 1999; Ingley, 2012). In our model, Lyn can phosphorylate other receptors on the same aggregate and also be trans-phosphorylated by other Lyn molecules, provided they all reside on the same aggregate. We simulate this relationship by providing a receptor bound Lyn access to receptors for phosphorylation that are one bond over on each side on an aggregate. This would indicate that as the size of an aggregate increases, we expect more Lyn to be associated with that aggregate, followed by more phosphorylation events. Since the sizes of the oligomers formed in the two different cell lines differed, we wanted to investigate the effect of aggregate sizes on ITAM phosphorylation. Figure 4.3 (A-D) display the amount of Igα and Igβ phosphorylation observed in the 697 and Nalm6 cell lines. We found that the presence of domains had an impact on the amount of receptor phosphorylation in both the cell lines (Figure 4.3 and Figure. 4.4). There were overall more phosphorylated ITAMS in 697 cell line with domains as compared to no domains (Figure 4.3 A,B and Figure 4.4). Similar trends were seen when comparing Nalm6 cell line with domains and without domains (Figure 4.3 C, D and Figure 4.4).

It was also interesting to note that the amount of single phosphorylated Igα and Igβ were at similar levels in both the cell lines, however, there were differences in the amount of doubly phosphorylated ITAMs (Figure 4.4). The amount of doubly phosphorylated Igα ITAMS were considerably higher in the Nalm6 cell line with domains as compared to 697 (Figure 4.4).

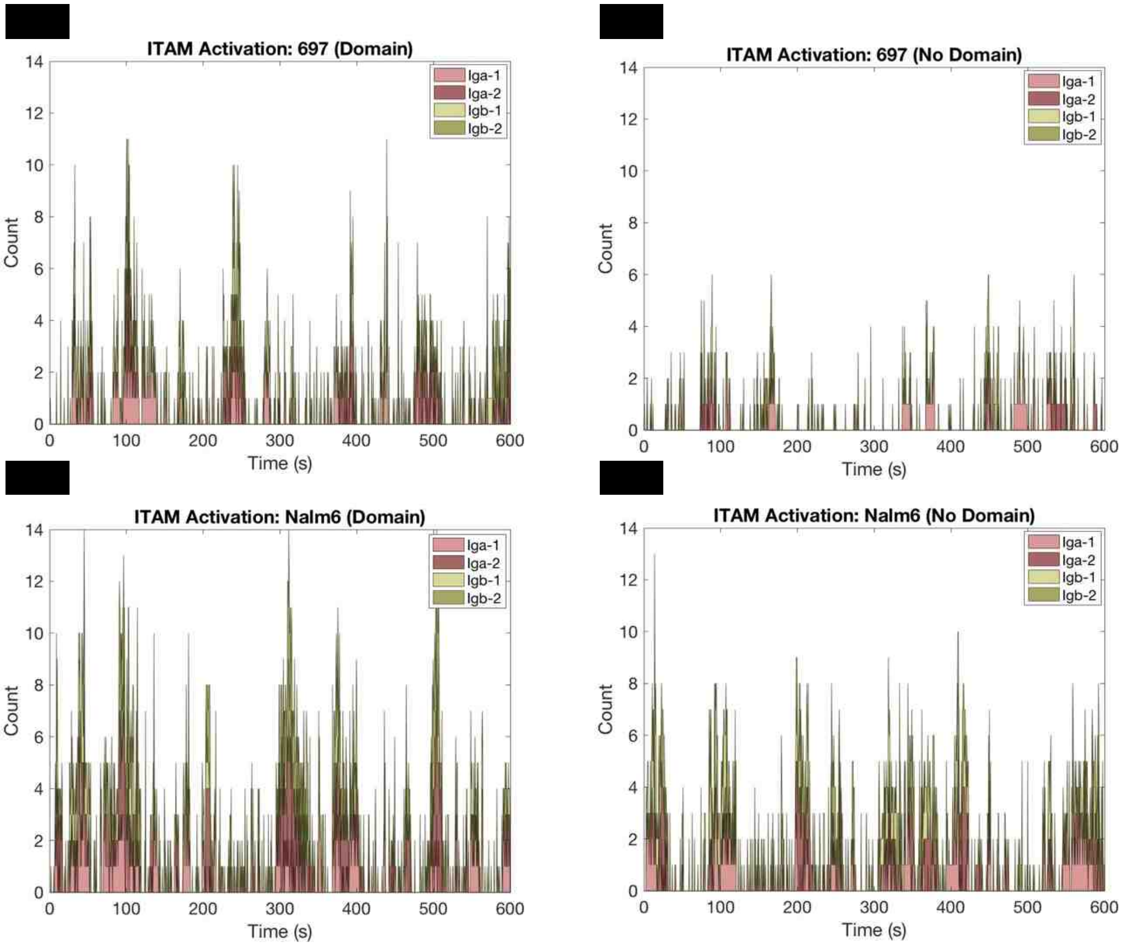


Figure 4.3: Receptor Igα and Igβ phosphorylation on 697 and Nalm6 cell line. (A) 697, Domain. (B) 697, No Domain. (C) Nalm6, Domain. (D) Nalm6, No Domain. Iga-1: once phosphorylated, Iga-2: twice phosphorylated, Igb-1: once phosphorylated, Igb-2: twice phosphorylated.

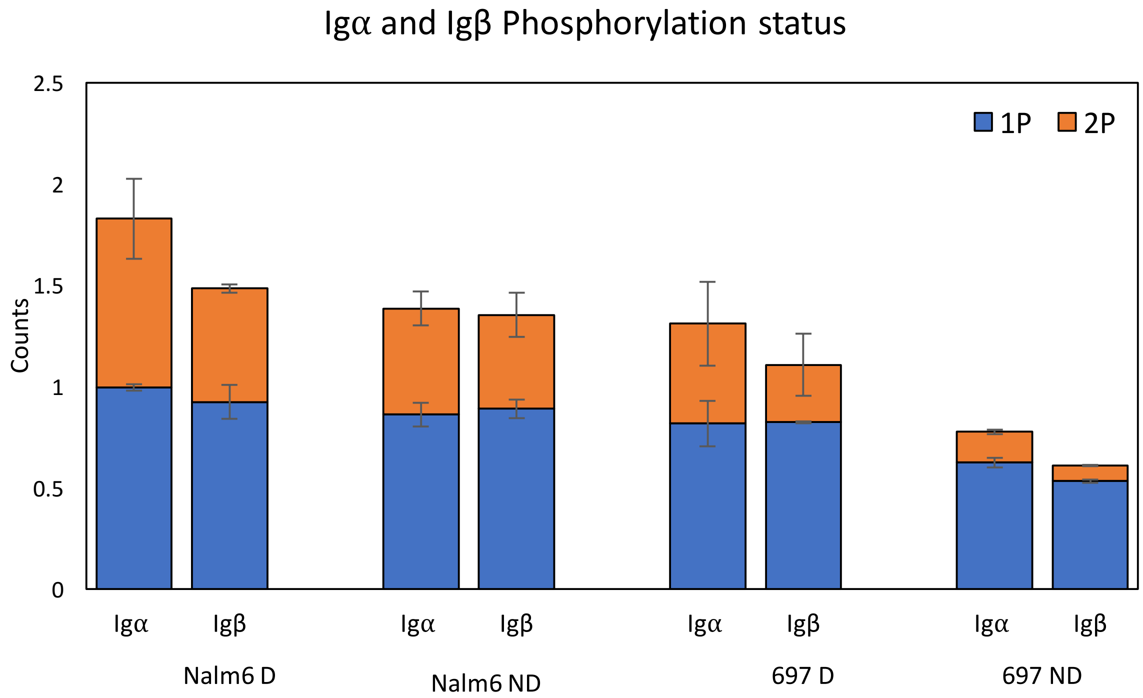


Figure 4.4: ITAM (Igα and Igβ) phosphorylation status of 697 and Nalm6 cell lines (with and without domains). All stacked bars are averages of 2 runs (± standard deviation) between 10 and 600 seconds. Nalm6D: Nalm6 with domains; Nalm6 ND: Nalm6 with no domain; 697 D: 697 with domains; 697 ND: 697 no domains.

4.4.3 Impact of varying dimer off rate and domains on Lyn binding and phosphorylation

Lyn molecules can bind to receptors through their unique or SH2 domains (Boggon and Eck, 2004). Receptor bound Lyn can trans-phosphorylate other Lyn molecules and thus, activate other Lyn kinases (Barua et al., 2012). This causes an increase in the catalytic activity of Lyn when the Lyn gets phosphorylated in its activation loop (Ingley, 2012). In the model, Lyn can bind to the ITAMS on the pre-BCR depending on the phosphorylation status of the ITAMs. Lyn can bind to unphosphorylated Igα through its unique domain or phosphorylated Igα through its SH2 domain. Lyn can also bind to Igβ through its SH2 domain. Once bound, Lyn molecules can phosphorylate the ITAMS as well as other Lyn molecules, one receptor over, on the aggregate. Once Lyn molecules are phosphorylated in their activation loop tyrosine site, they are assumed to be activated kinases. We wanted to investigate whether the different dimer off rates between the cell lines and the presence of domains have an effect on the amount of Lyn recruited to the receptors as well as their phosphorylation/activation status. From our simulations, we observed that the average amount of Lyn molecules recruited to the receptors were the same in three of the simulation conditions- Nalm6 with domains, Nalm6 without domains and 697 with domains (Figure 4.5 and Figure 4.6A). However, the amount of Lyn activation was considerably higher in Nalm6 with domains, followed by Nalm6 without domains, 697 with domains and 697 without domains (Figure 4.6B). Clearly, the different dimer off rate and the presence of domains were affecting the phosphorylation status of the Lyn molecules.

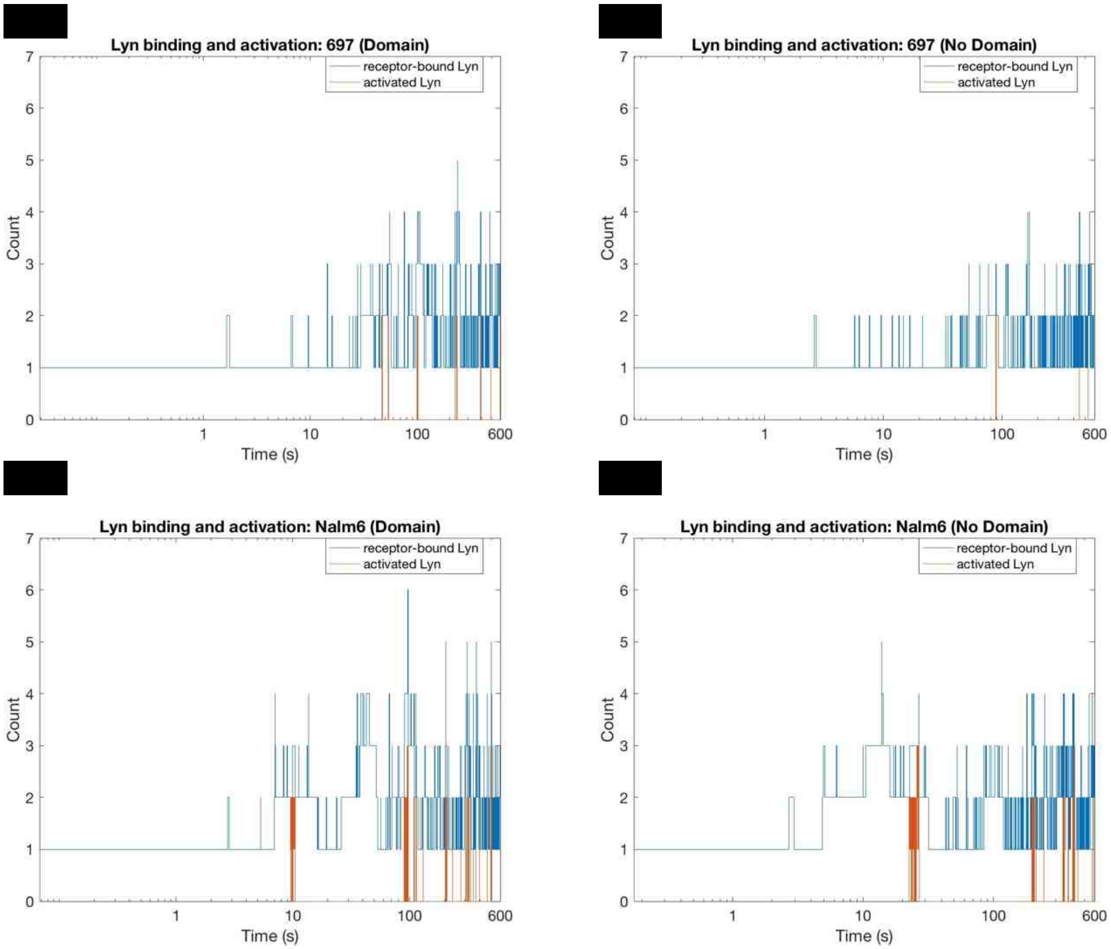


Figure 4.5: Receptor bound Lyn and activated on 697 and Nalm6 cell line. (A) 697, Domain. (B) 697, No Domain. (C) Nalm6, Domain. (D) Nalm6, No Domain.

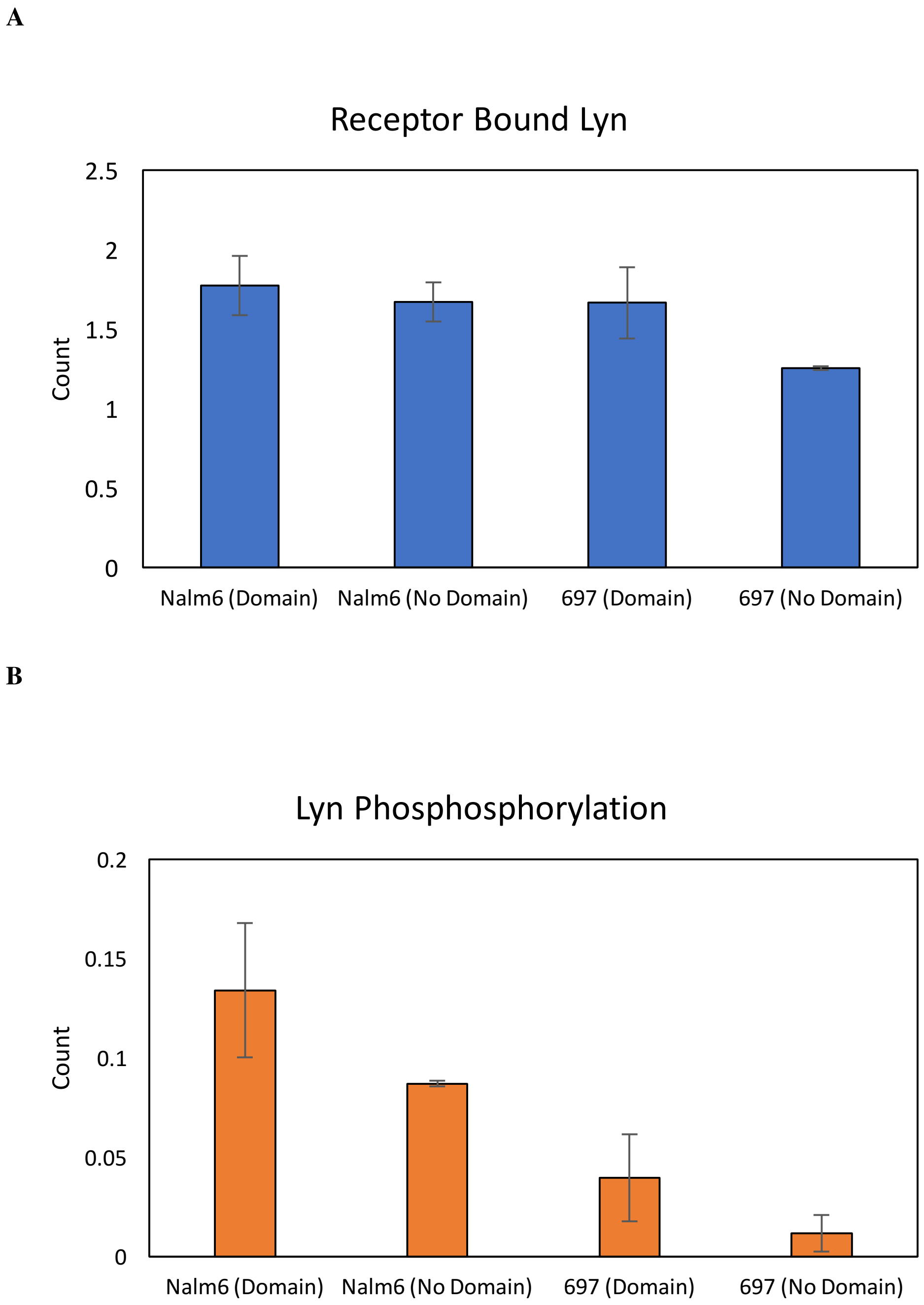


Figure 4.6. Lyn bound and Lyn phosphorylation counts. (A) Amount of receptor bound Lyn. (B) Amount of Lyn phosphorylated. All bars are the averages of 2 runs (± standard deviation) between 10 and 600 seconds.

4.4.4 Impact of varying dimer off rate and domains on Syk binding

The phosphorylation of ITAMs by Lyn molecules leads to creation of docking sites for SH2 domain containing Syk molecules (Kurosaki et al., 1995). Syk can bind to both singly or doubly phosphorylated Igα and Igβ, as they have two SH2 domains. Binding to doubly phosphorylated ITAMs is stronger than binding to singly phosphorylated ITAMs (Tsang et al., 2008). Binding is also affected by the preference of Syk molecules for Igα over Igβ (Kurosaki et al., 1995). Once receptor bound, Syk molecules can be phosphorylated by other Lyn molecules in its linker regions or it can be phosphorylated by other Syk molecules in its catalytic domain (Keshvara et al., 1998). Syk is assumed to be an activated kinase upon phosphorylation of its catalytic domain(Keshvara et al., 1998). We wanted to investigate the effect of different dimer off rates and presence of domain on Syk associations with the receptor as well as Syk phosphorylation status. Apart from the differences in the dimer off rate between the two cell lines we also know from SPT and experimental measurements that the amount of Syk molecules varies markedly in the two different cell lines with Nalm6 having a much higher amount of Syk molecules. Our simulations accordingly predict higher levels of receptor bound Syk in the Nalm6 line, as compared to 697 cell line (Figure 4.7A,D and Figure 4.8A). Cell lines with domains also had higher levels of Syk bound to the receptor (Figure 4.7 and Figure 4.8A).

We did not observe phosphorylation in the catalytic domain of Syk by other Syk molecules (Figure 4.7) in our simulations. We did ,however, find Syk phosphorylated in its linker regions by Lyn (Figure 4.7 and Figure 4.8B). The amount of Syk phosphorylation was the highest in Nalm6 cell line with domains, followed by Nalm6 without domains, 697 with domains and 697 without domains (Figure 4.8B).

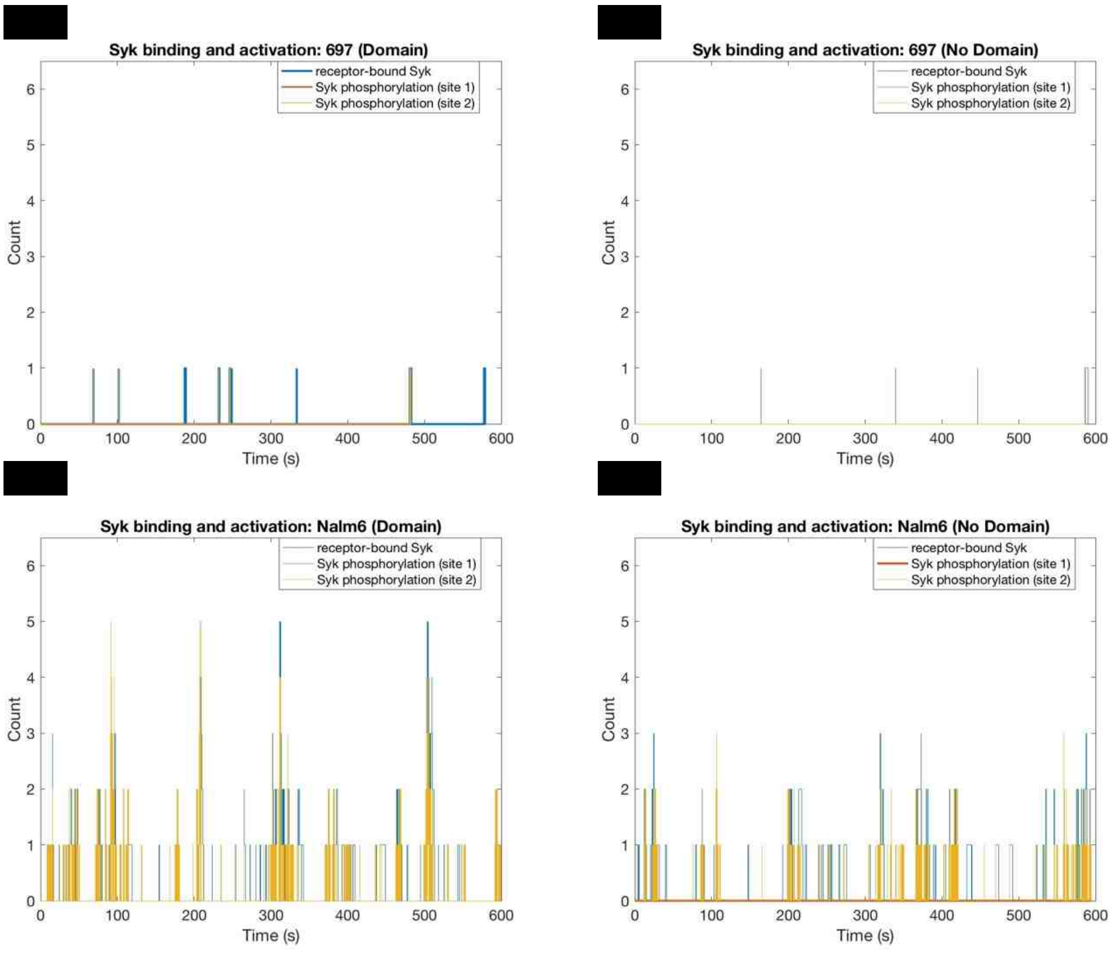

[redacted]
[redacted]
[redacted]
[redacted]

Figure 4.7: Receptor bound Syk on 697 and Nalm6 cell line. (A) 697, Domain. (B) 697, No Domain. (C) Nalm6, Domain. (D) Nalm6, No Domain. Syk phosphorylation (site 1): Syk phosphorylated by other Syk molecules in the catalytic domain. Syk phosphorylation (site 2): Syk phosphorylated by other Lyn molecules in the linker region.

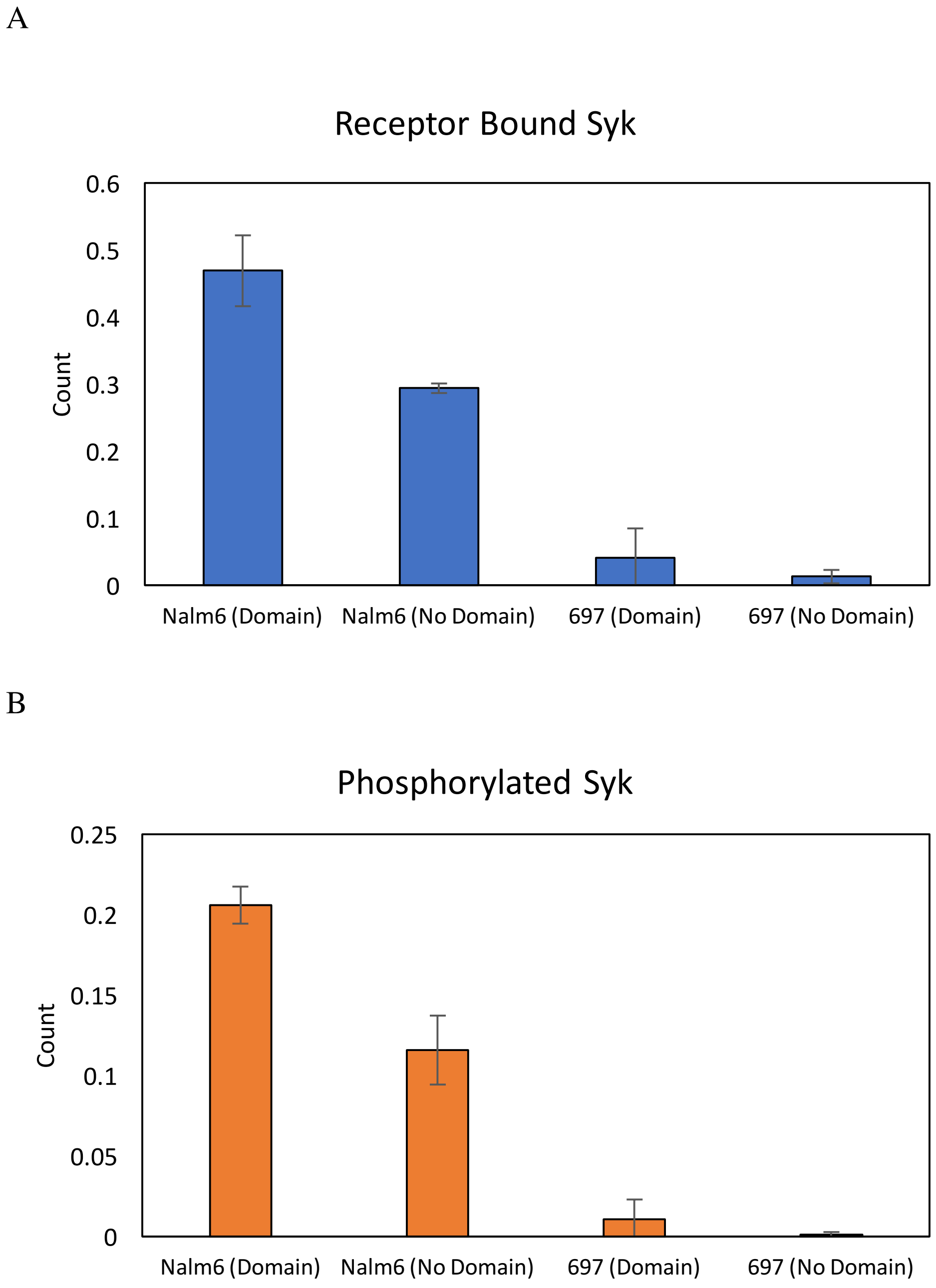


Fig 4.8. Syk bound and Syk phosphorylation counts. (A) Amount of receptor bound Syk. (B) Amount of Syk phosphorylated by Lyn. All bars are the averages of 2 runs (± standard deviation) between 10 and 600 seconds

4.5 DISCUSSION

In this study, we developed a spatial stochastic model to explore tonic signaling between the two different cell lines (697 and Nalm6) of leukemic pre-BCR. We found differences in sizes of the aggregates and molecule phosphorylation levels based on the different dimer off rates and presence of domains. The presence of domains impacted the pre-BCR chain lengths with domains favoring formation of higher order oligomers (Figure 4.2). The different dimer off rates between the two cell lines also had a major impact on the size of the aggregates formed (Figure 4.2). The Nalm6 cell line with the lower dimer off rate increased receptor aggregate sizes. From SPT measurements, we had speculated that Nalm6 was forming larger aggregates as dimer pairs that were more than 100 nm were often observed. Here, we provide strong evidence that the lower dimer off rate as measured in Nalm6 impacts the aggregate sizes in the pre-BCR leukemic cell line.

We also investigated receptor phosphorylation levels in these receptors. The overall phosphorylation level of receptors was found to be very low with the highest amount of phosphorylation seen in Nalm6 with domains. About 2.6% of Igα and 2.1% of Igβ were found to be phosphorylated in these simulations. In the 697 cell line, we found about 1.8% of Igα and 1.6% of Igβ phosphorylated. Thus, these numbers indicate that tonic signaling entails very low level of receptor phosphorylation. Low levels of receptor phosphorylation mean that the amount of important downstream signaling molecules such as Syk recruited to the receptor would also be very low since Syk docks to phosphorylated ITAMs. The amount of receptor phosphorylated might set up an upper limit for signaling activation of this pathway.

We also investigated the amount of Lyn molecules recruited to the pre-BCRs from the pool of available Lyns. The amount of receptor bound Lyn was seen to be at similar levels in Nalm6 with and without domains and 697 with domains. This amount varied between 25% and 23% in these three conditions. The lowest amount of receptor bound Lyns was seen in simulations of 697 with no domains. Here, the amount of Lyn bound was found to be about 18%. Thus, in tonic signaling, about a quarter percent of the pool of available Lyns seem to be receptor bound at steady state in both the cell lines. The total Lyn phosphorylation levels were also very low with about 2% of Lyn phosphorylated in Nalm6 with domains and 0.56% of Lyn phosphorylated in 697 with domains. Thus, activated Lyn was also present at a very low percentage during simulation of tonic signaling. This indicates that tight control of Lyn recruitment is implemented in these cells to keep subsequent receptor and Lyn phosphorylations at low levels.

The total amount of receptor bound Syk in Nalm6 with domains was 0.05% and it was 0.02% in 697 with domains. Syk receptor phosphorylation in the linker region by Lyn was also very low. About 0.02% of Syk molecules were phosphorylated on this site in Nalm6 with domains and 0.006% were phosphorylated in 697 with domains. We did not observe phosphorylation of Syk in its catalytic domain by Syk. Thus, recruitment along with phosphorylation of Syk molecules appears to be at very low levels in these cells.

Thus, from the simulations conducted in this study, it has become apparent that during tonic signaling, even though a large number of receptors might be involved in forming higher order aggregates, the total amount of Lyn and Syk molecules bound to these receptors remain low. The receptor, Lyn and Syk phosphorylation were also observed to be at very low levels. A reason for such low level of signaling could be that pre-BCRs have to judiciously regulate their signaling during their early development as too little or too much signaling could lead to apoptosis and impediment of their development pathway (Erasmus et al., 2016).

Erasmus *et al*., also obtained B cell progenitor cells from two BCP-ALL patients (patient# 238 and patient# 280) that were positive for the pre-BCR. Remarkably, pre-BCR on the surfaces of cells from the two patient samples displayed diffusion characteristics that were similar either to 697 cells (patient# 238) or to Nalm6 cells (patient# 280). Cells from patient# 280 were found to have slightly higher levels of BCL6 than from patient# 238 and were also more sensitive to antibodies against the VrepB region of the SLC which blocked pre-BCR dimerization. From experimental measurements, we know that Nalm6 has a higher number of Syk molecules. From the spatial stochastic model, we can observe that the Nalm6 cell line also forms higher order oligomers as well as more Lyn and Syk recruitment and phosphorylation. Thus, this could provide an explanation for the presence of higher amount of BCL6 in the patient pre-BCR cell line that behaved like Nalm6. The higher order oligomers in Nalm6, combined with an increased density of Syk molecules in the cell, led to increased BCL6 production.

This model can be used to further test knockouts of different protein for observing their effect on this signaling pathway. Obtaining the parameters for this pathway had proven to be very challenging. More experimental measurements are needed, so that the model can be fully biologically validated. We present here a model of basic tonic pre-

BCR aggregation and signaling. Other acting proteins in this pathway can be added in the future as more parameters become available.

CHAPTER 5: DISCUSSION

5.1 SUMMARY

In this dissertation, we explored the heterogeneity that exists across different biological scales using a variety of mathematical and computational methods. We used both deterministic (differential equations based) and stochastic methods to build models that were populated with both patient and experimental data in order to make biologically relevant predictions. In this first section of the discussion, a summary of the biological insights gained through the use of these models is presented.

In chapter 2 of this dissertation, we explored the heterogeneity that exists in the tumor microenvironment and its impact on patient therapeutic outcomes using a mathematical model of drug transport. This model considered patient specific parameters such as the blood vessel perfusion and radius of blood vessels, which tends to vary between patients, for predicting the effect of chemotherapy on patient tumors. We used H&E stained histological cross sections from patients as well as data from their CT scans to populate our model for making patient specific predictions. We found that patients who exhibited higher blood vessel perfusion in their tumors, also displayed a better prognosis in their treatment outcomes. This was because in well perfused tumors, drugs could easily reach the cancer cells whereas in tumors, where the blood vessel perfusion was low, cancer cells had a higher chance of escaping the toxic drugs. This study highlighted that even in patients who have the same type of cancer, treatment outcomes can very because each patient's tumor microenvironment is differently shaped and formed. Not only are genetic and cellular markers necessary for making decisions regarding patient treatment, but a patient's overall tumor microenvironment also needs to be taken into consideration before any therapy is administered.

In chapter 3 of this dissertation, we investigated the role of membrane domains in regulating cell signaling emanating from the ErbB2/ErbB3 receptor dimer. ErbB2 and ErbB3 receptors have been found to be overexpressed in many cancers and together form a very potent signaling unit. Since, unnecessary signaling from this receptor could be highly deleterious for a cell, we wanted to investigate how these receptors might be regulated by the membrane domains on the cell's surface. We found that the amount of signaling from these receptors was dependent on the degree of overlap between their domains. Additionally, we also found that increasing the strength of the confinement of receptors in domains only affected signaling when the receptors were completely segregated. In essence, the domains tightly regulated the receptors' proximity with each other to control cell signaling events.

The pre-BCR appears at a critical junction in the development of B lymphocytes. This is where the progenitor B cells decide whether to undergo apoptosis or to continue to develop into a mature B cell. A subset of patients with B-cell acute lymphoblastic leukemia also show expression of this receptor. In these patients, the leukemic cells exploit the tonic signaling pathway to survive and proliferate indefinitely. Therefore, understanding the regulation of this receptor is essential to devise strategies to combat this cancer. We investigated tonic signaling emanating from this receptor using two different BCP-ALL cell lines. In chapter 4, we created a spatial stochastic model of pre-BCR aggregation and populated the model with data acquired through SPT. We found differences in receptor aggregation and downstream signaling events based on the characteristic difference between the two cell lines.

5.2 SIGNIFICANCE

Our group has been involved in building lattice based and lattice free models of receptor signaling pathways with a focus on investigating the impact of spatial heterogeneity on membrane surfaces (Mayawala et al., 2005b; a; 2006; Hsieh et al., 2008; Costa et al., 2009; Hsieh et al., 2010; Costa et al., 2011; Pryor et al., 2013; Pryor et al., 2015). In chapter 3 of this dissertation, we used and extended the 2-D spatial stochastic model developed by Pryor *et al*., to investigate varying domain overlaps and receptor confinement on ErbB2/ErbB3 signaling pathways (Pryor et al., 2015; Kerketta et al., 2016). In the previous study, static spatial data gathered from SPT was used to create confinement zones for receptors which represented membrane domains in which receptors were "trapped" and had to pay a penalty for escape. This gave rise to spatial inhomogeneity on the membrane surface by creating dense or sparse areas of receptor population. We used the same principles in chapter 4 to build receptor domains from static spatial data of pre-BCR that were specific for this cell type. These receptor domains were used in conjunction with a 3-D spatial stochastic model of pre-BCR that simulated receptor diffusion, aggregation and phosphorylation, thus shedding insight on tonic signaling associated with this receptor.

5.2.1 Lattice free model of immunoreceptor signaling

Lattice based stochastic simulations, which take into account the spatial effect of receptor diffusion by displacing the molecules on a grid like membrane, have been used to investigate signaling in the mature B cell receptor (BCR) (Tsourkas et al., 2012; Mukherjee et al., 2013). However, spatial inhomogeneity existing on a biological membrane cannot be accurately represented by grid like confinement domains as these domains are dynamic with respect to time and are highly irregular in their shape and size. Hence, a more accurate representation of these domains can be built through reconstruction of some of the membrane domains observed during SPT. The pre-BCR spatial stochastic model presented in chapter 4 is a lattice free model, with receptor confinement zones recreated directly from experimental data, and hence might more accurately represent the spatial inhomogeneity present on the membrane surface of the immunoreceptors. Thus, this model can be used as a platform to simulate spatial inhomogeneity in other immunoreceptors including the BCR and T cell receptor (TCR).

5.2.2 The 3-D spatial stochastic model

The need for developing a 3-D model arose from the need to investigate the effect of spatial inhomogeneity on important cytoplasmic signaling molecules such as Syk in the pre-BCR pathway. Syk diffuses in the cytoplasm and transduces signal downstream of the pre-BCR. Thus, in order to capture the diffusion and reaction kinetics of Syk and its impact on the pre-BCR signaling, a 3-D spatial stochastic model had to be developed. This model holds the potential to simulate different cytoplasmic molecules important in either the pre-BCR pathway or other receptor signaling pathways. Therefore, this model provides a platform for developing 3-D spatial stochastic models of other important receptor pathways.

5.2.3 Aggregate sizes of immunoreceptor chains are directly obtained from the model

Although SPT of receptors is extremely useful in generating data on the diffusional and reaction dynamics of receptors on a live cell membrane, the low level of receptor labeling renders some of the information such as sizes of higher order oligomers inaccessible. In order to gain access to such information, computational models, that simulate receptor reactions using basic parameters obtained through experimental measurements, can be used to extrapolate the aggregate sizes formed in such receptor systems. The pre-BCR spatial stochastic model presented here, uses basic parameters obtained from SPT and experimental data to model formation of different sized aggregates and present the oligomer size as a key output of the model. This is in contrast to recent modeling efforts investigating the B cell receptor (BCR), where aggregation of receptors was modeled implicitly and the actual oligomer sizes were unreported (Barua et al., 2012; Mukherjee et al., 2013). Hence, this model can be used to simulate receptors which undergo aggregation and where reporting of the distinct oligomer sizes is a key requirement of the model simulations.

5.3 FUTURE INVESTIGATIONS

As mentioned in the above sections the 3-D spatial stochastic model of pre-BCR can be used to explore other immunoreceptors that undergo aggregation. Moreover, other cytoplasmic molecules can be added to further refine the model. Below is a description of potential future explorations in the pre-BCR signaling pathway.

5.3.1 Addition of Lyn specific domains on the membrane

One of the avenues that can be investigated is the effect of Lyn domains on the pre-BCR signaling pathway. Since Lyn is a membrane bound molecule, there is likely to be existence of Lyn rich domains on the membrane. It would be interesting to explore the effect of both Lyn and receptor specific domains on the signaling pathway. However, in order to reproduce Lyn specific domains on the *in silico* membrane, SPT with quantum dots tagged to the membrane bound Lyn molecule will have to be utilized. This might be experimentally challenging, however, if this objective is achieved then it would further shed light on the recruitment of Lyn molecules from Lyn specific domains to pre-BCR specific domains upon receptor aggregation.

5.3.2 Model parameter calibration with further experimental measurements

For the pre-BCR model, we relied heavily on data present in the literature for parameters such as phosphorylation and dephosphorylation rates for the receptors, Lyn and Syk molecules. To enable more accurate representation of tonic signaling, these model parameters need to be measured directly in the pre-BCR cell lines used in the experimental study. The model then needs to be recalibrated with the updated rates for more precise estimations of tonic signaling events.

APPENDICES

APPENDIX A: CHAPTER 3 SUPPLEMENT

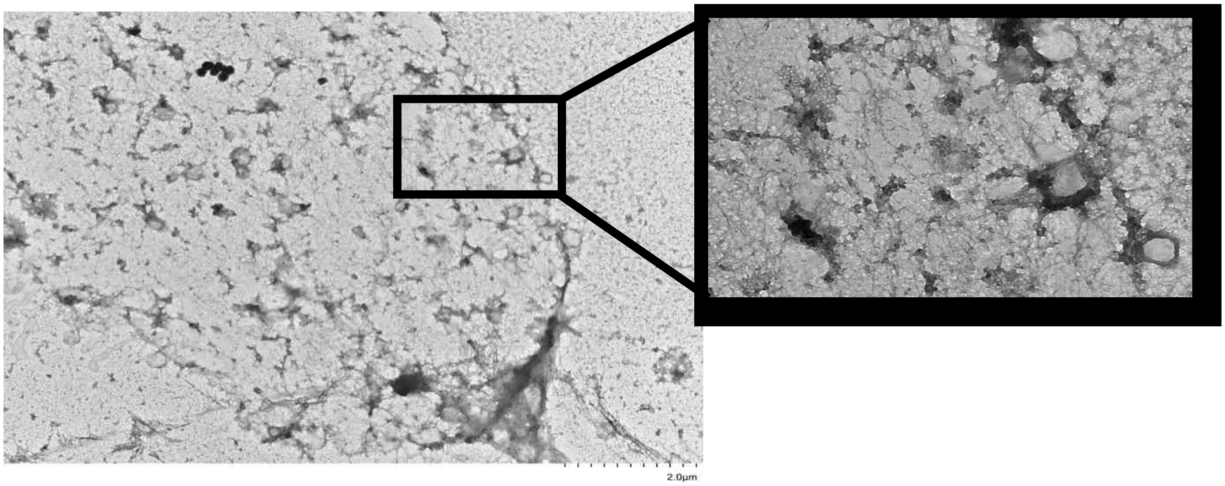


Figure A.1: Electronmicroscopy image of pre-b cell

APPENDIX B: RANDOM 3-D SIMULATION SPACE GENERATOR

```
close all; clear all; clc
datafilename = '08_25_17_Nalm6Run8' % save the simulation file with this
name'03_07_17b
%Receptor information
%[pBCR Lyn Syk] % Lyn
NP_rec = [71 7 959]; %  [697 Nalm6] % if z is 1[syk338 #syk1918] % if z is 0.5[syk169
% syk959]
numdomain= [0]   % domains,
%For 2-D
%NP_rec = [795]
NP = sum(NP_rec)
%Membrane information (in micrometer)
xlimmin = 0
xlimmax = 1.5
ylimmin = 0
ylimmax = 1.5
zlimmax = 0
zlimmin = -0.5
Membranevolume = xlimmax*ylimmax*abs(zlimmax)
% For 2_D
%MembraneArea = xlimmax*ylimmax
point_x = xlimmin + (xlimmax-xlimmin)*rand(NP,1)
point_y = ylimmin + (ylimmax-ylimmin)*rand(NP,1)
point_z = zlimmax + (zlimmin-zlimmax)*rand(NP_rec(3),1)

pBCR = [point_x(1:NP_rec(1)) point_y(1:NP_rec(1)) zeros(NP_rec(1),1)
zeros(NP_rec(1),1) ones(NP_rec(1),1)]
Boss = (ones*(1:NP_rec(1)))'
r_pBCR = [pBCR Boss zeros(NP_rec(1),1) zeros(NP_rec(1),1)]

np   = NP_rec(1)+NP_rec(2)
Lyn  = [point_x(NP_rec(1)+1:np) point_y(NP_rec(1)+1:np) zeros(NP_rec(2),1)
zeros(NP_rec(2),1) ]
Syk  = [point_x(np+1:NP(end)) point_y(np+1:NP(end)) point_z zeros(NP_rec(3),1)]

pBCRdomainRec = 0

%r_Lyn =  [Lyn 2*ones(NP_rec(2),1)]
%r_Syk =  [Syk 3*ones(NP_rec(3),1)]

figure(1)
%plot(pBCR(:,1),pBCR(:,2),'or')
```

```
plot3(pBCR(:,1),pBCR(:,2), pBCR(:,3),'ro', Lyn(:,1),Lyn(:,2),Lyn(:,3),'b*',
Syk(:,1),Syk(:,2),Syk(:,3),'go')
box on
xlabel('\mum, x','FontSize',20)
ylabel('\mum, y','FontSize',20)
zlabel('\mum, z','FontSize',20)
title('Simulation space','FontSize',18)
set(gca,'FontSize',10)
hold on
% set(gca,'xtick',linspace(xlimmin,xlimmax,3))
% set(gca,'XTickLabel',linspace(xlimmin,xlimmax,3))
%  set(gca,'ytick',linspace(ylimmin,ylimmax,3))
%  set(gca,'YTickLabel',linspace(ylimmin,ylimmax,3))
%  set(gca,'ztick',linspace(zlimmin,zlimmax,3))
%  set(gca,'ZTickLabel',linspace(zlimmin,zlimmax,3))
%set(gcf,'Position',[967   573   527   773])
%axis([xlimmin xlimmax ylimmin ylimmax zlimmin zlimmax])
 save (datafilename)

 %DIFFUSION


 % read the domains

FIN=fopen('ContourInfo.txt','r');

NContour = fscanf(FIN,'%d',1);


Contour = cell(NContour,1);

CSize = fscanf(FIN,'%d',NContour);

for iContour=1:NContour

    Contour{iContour} =fscanf(FIN,'%f',[CSize(iContour), 2]);

    Contour{iContour} = Contour{iContour} - repmat([15, 27.5],CSize(iContour),1);

end

fclose(FIN);

%figure(1);
%clf
```

```
MyColors = rand(5,3);

for iContour=1:5

plot(Contour{iContour}([1:end 1],1),Contour{iContour}([1:end 1],2),...
    'Color',MyColors(iContour,:));
end

xlim([-0.2 1.7]);
ylim([-0.2 1.7]);
axis equal

legend('Pre-BCR','Lyn','Syk','Domain 1','2','3','4','5');

plot([0 1 1 0 0]*1.5,[0 0 1 1 0]*1.5,'k-');
```

APPENDIX C: SCRIPT TO GENERATE INPUT FILES

```
%%
% Load data file
load 08_25_17_Nalm6Run8
%%
% Number of runs desired
q=8
StartFileNum=q;
EndFileNum=q;
%%
% Simulation Length [=] s
t=600;
%%
% Simulation time step
dt=0.00001;
% Print Frequency
printfreq=20;
N=t/dt;
%%
time=clock;
if time(2) < 10
   savedir=strcat('',['0' num2str(time(2))],'_',num2str(time(3)),'_',num2str(time(1)-
2000),'/')
elseif time(3) < 10
   savedir=strcat('',num2str(time(2)),'_',['0' num2str(time(3))],'_',num2str(time(1)-
2000),'/')
elseif time(2) < 10 && time(3) <10
   savedir=strcat('',['0' num2str(time(2))],['0' num2str(time(3))],'_',num2str(time(1)-
2000),'/')
else
   savedir=strcat('',num2str(time(2)),'_',num2str(time(3)),'_',num2str(time(1)-2000),'/')
end
mkdir(savedir)
%%
for jj = StartFileNum:EndFileNum
    numrun=strcat('Run_',num2str(jj),'/');
   if exist(strcat(savedir,numrun),'dir') == 0
      mkdir(strcat(savedir,numrun))
   end
    fid = fopen(strcat(savedir,numrun,'BMIP'), 'wt');
      fprintf(fid,'.\n');
      fprintf(fid,'%10.0f%10.0f%10.0f    # of
Particles\n',NP_rec(1),NP_rec(2),NP_rec(3));
      fprintf(fid,'%10.7f     Time Step [s]\n',dt);
```

```
        fprintf(fid,'%7.4f%7.4f%7.4f%7.4f%7.4f%7.4f     Membrane
Boundaries\n',xlimmax,ylimmax,xlimmin,ylimmin,zlimmin,zlimmax);
        fprintf(fid,'%10.0f     Data Print Frequency\n',printfreq);
        fprintf(fid,'%6.2f%3.0f  Length of Simulation [s], # of Domains\n',t,numdomain);
        fclose(fid);
        fid2 = fopen(strcat(savedir,numrun,'InitialParticleLoc'),'wt');
         for j = 1:size(pBCR,1)
            fprintf(fid2,'%18.16f%18.16f%18.16f%2.0f%2.0f%3.0f%2.0f%2.0f\n',(r_pBCR(j,
1)),(r_pBCR(j,2)),(r_pBCR(j,3)),(r_pBCR(j,4)),(r_pBCR(j,5)),(r_pBCR(j,6)),(r_pBCR(j,
7)),(r_pBCR(j,8)))
         end
         fid3 = fopen(strcat(savedir,numrun,'InitialParticleLoc_lyn'),'wt');
         for j = 1:size(Lyn,1)
            fprintf(fid3,'%18.16f%18.16f%18.16f%2.0f\n',(Lyn(j,1)),(Lyn(j,2)),(Lyn(j,3)),(Ly
n(j,4)))
         end
         fid4 = fopen(strcat(savedir,numrun,'InitialParticleLoc_Syk'),'wt');
         for j = 1:size(Syk,1)
             fprintf(fid4,'%18.16f%18.16f%18.16f%2.0f\n',(Syk(j,1)),(Syk(j,2)),(Syk(j,3)),(Sy
k(j,4)))
         end
        mkdir(strcat(savedir,numrun,'Data_Files'))
end
```

APPENDIX D: PRE-BCER SPATIAL STOCHASTIC SIMULATION PROGRAM

```
MODULE ModelConstants

  ! Variables defined in a module are accessible to any unit that uses the module

  ! General Constants
  DOUBLE PRECISION, PARAMETER :: Pi = 3.14159265

  !*Diffusion-Reaction Model Parameters*!

  ! NOTE base units are:
  !     length -- um (micrometer 1 um = 10^-6 m)
  !     time   -- s (second)

  ! ** Diffusion **
  DOUBLE PRECISION, PARAMETER :: DiffCoeff_Monomer = 0.16    ! receptor
  DOUBLE PRECISION, PARAMETER :: EscapeProb = 0.2 !0.1; ! 0.0941  ! receptor
escape prob
  DOUBLE PRECISION, PARAMETER :: Syk_DiffCoeff_Monomer = 17  ! um^2/s ??
  DOUBLE PRECISION, PARAMETER :: Lyn_DiffCoeff_Monomer = 0.4 ! um^2/s
Stone et al. Nature Comm 2015

  ! ** Binding (dimerization)
  ! NOTE UnbindRad_<...> are the Smoldyn-style unbinding radii used to separate the
products of dissociation; they are normally set to 5x the BR
  ! Rec-Rec
  DOUBLE PRECISION, PARAMETER :: BindRad_Dimer = 0.000215 ! (rec-rec) (use
0.000313  from e2/e3) and use sim area 25 nanometer square for 10. or 75 by 75 for 100
  DOUBLE PRECISION, PARAMETER :: UnbindRad_RestDimer = BindRad_Dimer *
5 ! UBR
  ! Lyn-Rec
  DOUBLE PRECISION, PARAMETER :: Lyn_BindRad_Dimer = 0.000228519!2.25e-4
! unique domain binding to iga for sh2 binding BR^2 is scaled up by [100 (iga) 20 (igb)]
!  (2D Smoldyn: 1.4e-2 /um^2)
   ! value of BR=1.34e-5 um  (dt=1e-5) obtained by trial and error from Smoldyn to
match the on-rate of 4.6e-5 /(s#/um^2)
  double precision, dimension(3), parameter :: LynBindScaleFactor = [1.0, 1.0, 0.5] ![1.0,
1.0, 0.5]! rel. rates [unique dom, Igalpha(Ph>0), Igbeta(Ph>0)]
  DOUBLE PRECISION, PARAMETER :: Lyn_UnbindRad_RestDimer =
Lyn_BindRad_Dimer * 5  ! the * 10 is for activated Ig sites; use it for all Lyn unbinding
  double precision, parameter :: Lyn_available_fraction = 1 ! 1! only 3.5% of (un-
activated) Lyn is in a state where it is able to bind receptor
  ! Syk-Rec
  DOUBLE PRECISION, PARAMETER :: Syk_BindRad_Dimer =0.00157424 !
represents Syk to Igb binding; others scaled in the code (see next line)
  ! double precision, dimension(2), parameter :: SykBindScaleFactor = [1.0, 3.0] ! relative
binding rates [Igalpha(Ph>0), Igbeta(Ph>0)]
  ! TODO change to this
   double precision, dimension(4), parameter :: SykBindScaleFactor = [1.0/12.0, 1.0,
1.0/36.0, 1.0/3.0] ! relative binding rates [Igalpha(Ph=1,2), Igbeta(Ph=1,2)]
  DOUBLE PRECISION, PARAMETER :: Syk_UnbindRad_RestDimer =
Syk_BindRad_Dimer * 5 ! UBR 0.0000111

  ! ** Dissociation (unbinding)
  DOUBLE PRECISION, PARAMETER :: Dimer_off_rate= 0.164! Rec-Rec ! "true
dimer lifetime" based rate from Adam - 0.772 /s (697) , 0.164 /s (Nalm6)
  DOUBLE PRECISION, dimension(2), PARAMETER :: Lyn_dimer_off_rate = [20.0,
0.12 ]! Lyn-Rec ! if Lyn_Site = 1, then first off rate, if Lyn_Site = 2, then second off rate
  DOUBLE PRECISION, dimension(2), PARAMETER :: Syk_dimer_off_rate = [ 2.6,
0.63]! Syk-Rec ! if Syk_Site = 1, then first off rate, if Syk_Site = 2, then second off rate

  ! ** Phosphorylation **
  ! (Receptor)
  ! Lyn mediated phos rates depend on (Lyn phos state (InActive,Active) x  substrate
ITAM state (P0,P1) )
  ! !! CAUTION !! The rates below have a factor of 10x or 100x for the inactive Lyn case
over the rate we estimated from Barua et al (4.93e-4 /s - 0.296 /s)
  !             a factor of ~3-10 is justified by the inverse (Rec-Rec bond count):(total Rec
count) ratio (i.e. only 10-30% of receptors are bound at all)
  !             the rate is much higher in another paper (0.5 /s in Weiss and 100 /s in
Tsourkas )
  double precision, dimension(4), parameter :: Phos_rate = [ 30,15,100,50 ] ! /s Lyn
mediated phos rates by substrate and Lyn state: [IA(P0), IA(P1), A(P0), A(P1)]
  double precision, dimension(2), parameter :: Phos_off_rate = [ 20, 40 ] ! receptor
dephos rate (/s) depends on initial state [P1,P2]
  !double precision, dimension(4), parameter :: Phos_rate = [
100*0.000985,100*0.000493,0.296,0.148 ] ! /s Lyn mediated phos rates by substrate and
Lyn state: [IA(P0), IA(P1), A(P0), A(P1)]
  !double precision, dimension(2), parameter :: Phos_off_rate = [ 1.0, 2.0 ] ! receptor
dephos rate (/s) depends on initial state [P1,P2]
  ! (Syk)
  DOUBLE PRECISION, dimension(2), PARAMETER :: Syk_Phos_Rate = [100, 200] !
[IA,A] Syk-mediated for Unactivated syk & activated syk
  !  DOUBLE PRECISION, dimension(3), PARAMETER :: Syk_Phos_Rate = [0.0148,
1.48, 0.5]! [IA,A,self] Syk-mediated for Unactivated syk & activated syk + syk phos by
itself
  ! TODO (?) eliminate the self entry from Syk_Phos_Rate
  ! TODO (?) introduce separate Syk phos by Lyn rate
  DOUBLE PRECISION, PARAMETER :: Syk_DePhos_Rate = 20.0 ! /s
```

```
 ! (Lyn)
 ! !! CAUTION !! The rates below have a factor of 10x or 100x for the inactive Lyn case
 DOUBLE PRECISION, dimension(2), PARAMETER :: Lyn_Phos_Rate = [30,100] !
[100*0.000493, 0.148] ! Lyn mediated [IA,A]
 DOUBLE PRECISION, PARAMETER :: Lyn_Dephos_Rate = 20.0 !  /s

END MODULE ModelConstants

MODULE ParticleInfo

 INTEGER, PARAMETER :: MaxAgg = 50; ! Largest expected aggregate (chain) length
 integer, parameter :: MaxContour = 500; ! largest expected number of points defining a
contour
 integer, parameter :: MaxDom = 20 ! maximum expected number of domains

 ! Put generally relevant variables here instead of in the Sytem_Information struct
 ! variables at-large relevant to the state of the system

 integer :: NumDomains,DomainParticleCount(MaxDom) ! number of domains and
particles in each of them

 !
 double precision, dimension(2) :: XBox, YBox, ZBox ! to replace
System_Info%SimSpace_Boundary
 integer :: Syk_Pick_Count=0, Syk_Intrinsic_Count=0, &
      Syk_PickNoReaction=0, Syk_FreePick=0, Syk_BoundPick=0,
Syk_BoundPick_Unbound=0, &
      Syk_DiffCall_Count=0, Syk_Diff_Reaction=0, Syk_Diff_NoReaction=0, &
      Syk_BindCall_Count=0, Syk_Bind_Reaction=0, Syk_Bind_NoReaction=0, &
      Enc_EligAggCount=0, Enc_BossCount=0
 integer, dimension(2) :: Enc_SysPhosCount
 double precision :: SysMinDist = 100

 double precision, parameter :: SykLayerDepth = 1.0e-1 ! thickness of layer close to the
membrane where Syk could possibly interact with membrane bound species

 ! Variable types specific to the simulation : domains, molecule types (receptor, lyn, syk)

 type Domain ! part of MODULE ParticleInfo

    integer :: ContourLength ; ! number of points in the contour (last point is the same as
the first)
    double precision :: Contour(MaxContour,2) ;! contour defining the domain
    ! double precision :: EscapeProb ! future; for now use a universal value
    double precision :: Xlim(2) ! max and min x coordinates for quick checking
    double precision :: Ylim(2) !
```

```
  end type Domain


  TYPE Lyn ! part of MODULE ParticleInfo

    DOUBLE PRECISION :: Position(3) ! (x1,y1)
    INTEGER :: Receptor_ID !This will be zero if the Lyn is free;
    INTEGER :: Phos ! 0= unactive, 1 = active
    INTEGER :: Itam_site ! 0 = unbound, 1 = Igalpha, 2 = Ig beta
    INTEGER :: Lyn_site  ! 0 = unbound, 1 = Unique, 2 = SH2

  END TYPE Lyn  ! part of MODULE ParticleInfo


  TYPE Syk ! part of MODULE ParticleInfo

    DOUBLE PRECISION :: Position(3) ! (x1,y1)
    INTEGER :: Receptor_ID ! !This will be zero if the Lyn is free
    INTEGER :: Phos !   0 = unactive, 1 = active (catalytic site is activated by adjacent Syk)
    INTEGER :: Phos_2 ! 0 = unactive, 1 = active (other phos site activated by adjacent Lyn)
    INTEGER :: Itam_site ! 0 = unbound, 1 = Igalpha, 2 = Igbeta
    INTEGER :: Syk_site  ! 0 = unbound, 1 = bound through 1 SH2 only, 2 = bound through 2 SH2s (tandem SH2s)

    ! TODO: check that Syk_site correctly reflects underlying SH2 state
    !       also check what happens if the ITAM site is phoshporylated AFTER Syk was bound
    ! TODO? Syk type could be merged with the Lyn type

  END TYPE Syk  ! part of MODULE ParticleInfo


  TYPE Molecule !  = Receptors ! part of MODULE ParticleInfo

    ! position - current
    DOUBLE PRECISION :: Position(3) ! (x1,y1,z1)
    DOUBLE PRECISION :: r_Squared ! r^2 of receptor calculated each pdt step

    DOUBLE PRECISION :: LastOnOffTime ! the time this particle got into the current aggregate configuration

    integer :: RecID  ! same as the index, useful for array manipulation
    integer :: Domain ! domain ID consistent with current position
    ! binding configuration
    INTEGER :: Bond        ! 0 = No bond, 1 = 1 bonds, 2 = 2 bonds
    INTEGER :: BoundRec_1 ! ID of receptor bound on site 1
    INTEGER :: BoundRec_2 ! --------------------------- 2

    ! containing aggregate
    INTEGER :: Boss          ! ID of the boss receptor of the containing aggregate
    INTEGER :: Agg_Size       ! size (number of receptors) of the containing aggregate

    ! itam state
    INTEGER :: Iga_Phos ! 0= unphosphorylated, 1 = singly phosphorylated, 2 = double
phosphorylated
    INTEGER :: Iga_Lyn  ! 0= no Lyn, some number is Lyn ID
    INTEGER :: Iga_Syk  ! 0= no Syk, some number is Syk ID
    INTEGER :: Igb_Phos ! 0= unphosphorylated, 1 = singly phosphorylated, 2 = double
phosphorylated
    INTEGER :: Igb_Lyn  ! 0= no Lyn, some number is Lyn ID
    INTEGER :: Igb_Syk  ! 0= no Syk, some number is Syk ID

  END TYPE Molecule  ! part of MODULE ParticleInfo

  TYPE SystemInformation ! part of MODULE ParticleInfo

    ! holds the current state of the system for handy access
    !
    ! TODO -- these global variables could simply be declared as such
    !        within the ParticleInfo module, I am not sure there is
    !        a need to keep them bundled like this

    CHARACTER(80) :: Save_Directory ! Parameter
    INTEGER :: Num_Particles ! Parameter , number of receptors -- **duplicated** by
Total_Rec_Count
    integer :: Num_Aggregates ! Variable, number aggregates of receptors
    INTEGER :: AggSizeCount(MaxAgg) ! Variable, keeps track of aggregate size
distribution
    DOUBLE PRECISION :: Time_Step !  Parameter
    DOUBLE PRECISION :: SimSpace_Boundary(6) !  Parameter
    INTEGER :: Print_Frequency ! Parameter,
    DOUBLE PRECISION :: Simulation_Time ! Parameter  (?)
    INTEGER :: Number_Domains ! Parameter
    DOUBLE PRECISION :: Current_Simulation_Time ! variable, system time
    LOGICAL :: Reaction  ! flag, indicates whether a reaction occurred in the latest
update
    INTEGER :: Total_Rec_Count, Total_Lyn_Count, Total_Syk_Count ! parameter
(number of spatial particles, bound or not)
```

```
    INTEGER :: Free_Lyn_Count, Free_Syk_Count ! global state variable
    INTEGER :: OutputLevel ! Parameter, switch

  END TYPE SystemInformation  ! part of MODULE ParticleInfo

  ! the state of the system is represented by instances of the above defined types
  ! part of MODULE ParticleInfo

  TYPE(Molecule), POINTER :: RecMolecule(:), RecMoleculeInitial(:),
RecMoleculePrevious(:)
  TYPE(Lyn),      POINTER :: LynMolecule(:), LynMoleculeInitial(:),
LynMoleculePrevious(:)
  TYPE(SYK),      POINTER :: SykMolecule(:), SykMoleculeInitial(:),
SykMoleculePrevious(:)
  type(Domain),   pointer :: Dom(:)
  TYPE(SystemInformation) :: System_Info

contains !  ! part of MODULE ParticleInfo

  function InDomain(Coord,DomID) result(Inside)

    implicit none

    real*8,    intent(in) :: Coord(2) ! (x,y) to test
    integer, intent(in) :: DomID    ! index of domain
    logical :: OnBoundary ! rarely this might be true

    logical :: Inside ! true if in the domain, false otherwise

    integer :: Counter(2) ! counts intersections (left,right)
    integer :: i1,i2 ! indices of the contour segment

    real*8 :: Xa,Xb,Ya,Yb,XP,YP, Xint

    ! NOTE: contours are assumed closed "by hand"
    !      i.e. we pretend the first point in the list FOLLOWS the last point
    !      but they SHOULD NOT be identical


    ! check if there is a domain by the index specified

    if (NumDomains < DomID) then
      write(*,*) ' InDomain: requested domain ',DomID,' does not exist'
      return
    end if
    ! algorithm idea:
    ! loop through the segments (12),(23),...(end-1 end),(end 1)
    ! for segment (ij) check if
    !    (a) YP between y(i), y(j)
    !    (b) if (a), is XP to the left of point where the (ij) segment intersects the horizontal
line y=YP
    !   --> count the times (a)(b) are true
    !   (XP,YP) is inside the  contour if and only if* the count is odd
    !
    ! * caveat -- if YP equals one or more of the y(k)'s, special procedure

    ! use these for clarity
    XP = Coord(1)
    YP = Coord(2)

    ! count the left (Xint < Xp) and right (xint > Xp) intersections
    Counter = 0

    ! just in case the point is exactly on the boundary
    OnBoundary = .false.

    ! default answer is outside
    Inside = .false.

    do i1=1,Dom(DomID)%ContourLength

      ! index of points in the contour segment
      i2 = i1+1
      if (i2>Dom(DomID)%ContourLength) i2=1

      ! xy of the two ends of the segment
      Xa = Dom(DomID)%Contour(i1,1)
      Xb = Dom(DomID)%Contour(i2,1)

      Ya = Dom(DomID)%Contour(i1,2)
      Yb = Dom(DomID)%Contour(i2,2)

      ! check for "YP between Ya,Yb
      if ( (Yb - YP) * (YP - Ya) > 0 ) then

        ! x coordinate of the intersection
        Xint = Xa + (Xb - Xa) * (YP - Ya) / (Yb - Ya)

        if(Xint < XP) then
          Counter(1) = Counter(1) + 1
        else if (Xint > XP) then
```

```
         Counter(2) = Counter(2) + 1
       else
         write(*,*) 'InDomain warning -- point ',XP,YP,' is on the boundary of domain
',DomID
         OnBoundary = .true.
       endif
       ! TODO: also figure out what to do when Ya=Yb or when
       ! the product (Yb - YP) * (YP - Ya)=0 (i.e. YP = Ya or Yb)

     end if

   end do

   if (modulo(Counter(1),2)==1) Inside = .true.

  end function InDomain ! part of MODULE ParticleInfo

END MODULE ParticleInfo

!! NOTE: Moved subroutines to the end of the file, preferably in the order of
dependencies
!!      -- i.e. main program first, then subroutines called by the program, etc.

PROGRAM Pre_BCR

  USE mtmod  ! used to generate random numbers

  USE ParticleInfo
  USE ModelConstants
  IMPLICIT NONE

  ! Declare variables local to the main program

  double precision :: DiffSTD, DiffSTD_Lyn, LynDiffSTD, DiffSTD_Syk!! diffusion
standard deviation

  DOUBLE PRECISION :: UnBindProb, LynBindProb, LynUnBindProb
  double precision :: PhosProb(2), DePhosProb(2)  ! used for receptors and also for Syk
(two sites, one act.by Lyn, one by Syk)
  DOUBLE PRECISION :: SykBindProb, SykUnbindProb ! SykTotalPhosProb
  double precision :: ProbVec(5), SumProb, Prob_1, Prob_2 ! used in choosing the
phosphorylation / dephos site
  integer :: PhosIndex, chosen_site                 ! used in choosing the phosphorylation /
dephos site
  integer :: PhosLevel              ! use to count phosphorylation of ITAMs
  !
  DOUBLE PRECISION :: r1, r2, x1, y1, z1, w1, w2, rannum, r3  ! Random Numbers
  INTEGER :: k, i, m, ii, w, lifecount, p, seed, seed_random(8), CurrentSpecies, next,size,
ic, d, f ! counters

  integer :: iBR, iITAM, BoundRecID, LynRecID, NeighborLynID, NeighborLynState,
NeighborSykState, NeighborSykID,SykRecID !

  INTEGER :: iLyn, iSyk, B1_Lyn, B2_Lyn
  INTEGER :: Boss, Bond_count_i,Bond_count_k, v, c,  NewBoss
  INTEGER :: NP, Lyn_num, Syk_num, NPT ! number of particles (receptors only /Lyn,
Syk / total)
  INTEGER :: printfreq, nd, domainnum, BoundRec_1, BoundRec_2! data print
frequency, number of frames, number of domains
  INTEGER :: Current_bond, Bond_Count! Used for select case switching
  DOUBLE PRECISION :: st, dt, t, pdt! time step, time, system time, timestep per particle
  DOUBLE PRECISION :: xlimmax, ylimmax,xlimmin,ylimmin, zlimmin, zlimmax !
width of simulation (x axis), length (y axis) of simulation
  DOUBLE PRECISION :: MSD ! MSD calculation
  CHARACTER*200 fnstring ! Filename string
  CHARACTER(80) :: outdir ! extra path info for HPC
  LOGICAL :: Reaction
  INTEGER*8 :: N, frames, datacut, moves, j,tt,o,y ! Number of moves, number of
frames, cycles until print is needed, total number of moves, move counter

  integer :: ParticleDomain(10)

!!!!!!! DO NOT USE FILE # 5 (DEFAULT INPUT FILE NUMBER) OR 6 (DEFAULT
OUTPUT FILE NUMBER) !!!!!!
  WRITE(*,*) 'FORTRAN Simulation Started'

  ! Open input files
  OPEN (1,file='BMIP')
  OPEN (2,file='InitialParticleLoc')
  OPEN (3,file='DomainLimits')
  OPEN (112,file = 'InitialParticleLoc_lyn')
  OPEN (13,file = 'InitialParticleLoc_Syk')

  open(17,file = 'ContourInfo.txt'); ! domain contours

  print *, 'Initializing:'

  print *, '    reading BMIP file..'

  ! Read in values from input file
  READ(1,107) outdir ! HPC Path info
  write(*,*) outdir
```

```
 READ(1,100) NP,Lyn_num,Syk_num ! # of particles - for now, receptors only --
TODO : input lyn, syk counts in BMIP
 write(*,*) NP,Lyn_num,Syk_num
 READ(1,101) dt ! Time step [s]
 READ(1,102) xlimmax,ylimmax,xlimmin,ylimmin, zlimmin, zlimmax ! simulation
boundaries
 ! write(*,*) 'Sim boundaries:',xlimmax,ylimmax,xlimmin,ylimmin, zlimmin, zlimmax
 READ(1,105) printfreq ! data print frequency
 READ(1,106) t, NumDomains ! simulation length [s], # of domains

 ! write(*,*) t, NumDomains

100 FORMAT(I10,I10,I10)
101 FORMAT(F10.7)
102 FORMAT(F7.4,F7.4,F7.4,F7.4,F7.4,F7.4)
105 FORMAT(I10)
106 FORMAT(F6.2,I3)
107 FORMAT(a)

 CLOSE(1)

 ! Set System Info
 System_Info%Save_Directory = outdir
 System_Info%Num_Particles = NP ! NP, "Particles" refers to receptors for now
 System_Info%Total_Rec_Count = NP
 System_Info%Num_Aggregates = NP
 System_Info%AggSizeCount = 0
 System_Info%AggSizeCount(1) = NP
 System_Info%Time_Step = dt
 System_Info%SimSpace_Boundary(1) = xlimmax
 System_Info%SimSpace_Boundary(2) = ylimmax
 System_Info%SimSpace_Boundary(3) = xlimmin
 System_Info%SimSpace_Boundary(4) = ylimmin
 System_Info%SimSpace_Boundary(5) = zlimmin
 System_Info%SimSpace_Boundary(6) = zlimmax
 write(*,*) 'Simulation Box boundaries: '
 write(*,*) 'x [',xlimmin,xlimmax,']'
 write(*,*) 'y [',ylimmin, ylimmax,']'
 write(*,*) 'z [',zlimmin, zlimmax,']'
 System_Info%Print_Frequency = printfreq
 System_Info%Simulation_Time = t
 System_Info%Current_Simulation_Time = 0
 System_Info%Reaction = .false.
 System_Info%Total_Lyn_Count = Lyn_num !1592 !
 System_Info%Total_Syk_Count = Syk_num !3844 !
 System_Info%Free_Lyn_Count = Lyn_num! 1592 !
 System_Info%Free_Syk_Count = Syk_num! 3844 !

 ! *** domain stuff ***

 allocate(Dom(NumDomains)) ! allocate memory for the required number of domains

 read(17,*) nd  ! number of domains in the contour file

 ! make sure there are enough contours
 if(nd < NumDomains) then
   write(*,*) 'Error: ', NumDomains, ' domains specified, found only ',nd
   return
 end if

 ! write(*,*) 'There are ',nd,' contours in the input file.'
 ! write(*,*) 'We are looking for ',NumDomains,' contours..'

 read(17,*) Dom(:)%ContourLength

 do i=1,NumDomains

   write(*,*) 'Domain ',i,' has ', Dom(i)%ContourLength, ' points.'

   read(17,*) Dom(i)%Contour(1:Dom(i)%ContourLength,1)
   read(17,*) Dom(i)%Contour(1:Dom(i)%ContourLength,2)

   ! shift the domains -- TODO take this out and put shited coordinates into a file
   Dom(i)%Contour(1:Dom(i)%ContourLength,1) =
Dom(i)%Contour(1:Dom(i)%ContourLength,1) - 15.0
   Dom(i)%Contour(1:Dom(i)%ContourLength,2) =
Dom(i)%Contour(1:Dom(i)%ContourLength,2) - 27.5


   Dom(i)%Xlim(1) = minval(Dom(i)%Contour(1:Dom(i)%ContourLength,1))
   Dom(i)%Xlim(2) = maxval(Dom(i)%Contour(1:Dom(i)%ContourLength,1))
   Dom(i)%Ylim(1) = minval(Dom(i)%Contour(1:Dom(i)%ContourLength,2))
   Dom(i)%Ylim(2) = maxval(Dom(i)%Contour(1:Dom(i)%ContourLength,2))

   write(*,*) '     bounds -- x:',Dom(i)%Xlim(:),' y:',Dom(i)%Ylim

!!$     do ii=1,Dom(i)%ContourLength
!!$        write(*,*) '     Dom ',i,' point
',ii,'x=',Dom(i)%Contour(ii,1),'y=',Dom(i)%Contour(ii,2)
!!$     end do

 end do
```

```
  close(17)



  System_Info%OutputLevel = 1
  ! Suggestion: 1 - one line per actual reaction, format for reading in matlab etc.
  !           2 - details eg. agg membership
  !           3 - debug stuff, what particle came in, intermediate steps etc

  ! *** end of inputs  ***

  ! Create output files
  open(4, file=TRIM(outdir)//'/TrueDimerLifeTimes') ! dimer lifetimes from the actual
simulation, not the frame rate
  open(7, file=TRIM(outdir)//'/MSDData')           ! MSD info written to according to
frame rate, calculated each dt
  open(8, file=TRIM(outdir)//'/TimeToPhos')        ! time to phosphorylation for each
dimer
  open(9, file=TRIM(outdir)//'/PhosLifetimes')      ! Phosphorylation time
  open(10,file=TRIM(outdir)//'//DomainExitInf')     ! Exit rate info

  ! Output file header lines
  WRITE(7,*) 'MSD' !
  WRITE(4,*) 'Reac  ',' Time Step  ','  i  ','  k  ','  AggSize  ','  AggSize  ','  AggSize  '

  ! Calculate number of moves
  N=t/dt

  ! we select from among all rec,lyn,syk, so this is what sets the effective time step
  NPT = System_Info%Total_Rec_Count + System_Info%Total_Lyn_Count +
System_Info%Total_Syk_Count

  ! Calculate number of data frames to record and store as an integer
  frames = INT(printfreq*t)           ! Number of frames to write out
  datacut = INT((N*NPT)/(printfreq*t))  ! Iterations (steps) one frame

  ! Give dimensions for the arrays and matrices
  ALLOCATE (RecMolecule(System_Info%Total_Rec_Count))
  ALLOCATE (LynMolecule(System_Info%Total_Lyn_Count))
  ALLOCATE (SykMolecule(System_Info%Total_Syk_Count))
  ALLOCATE (RecMoleculeInitial(System_Info%Total_Rec_Count))
  ALLOCATE (LynMoleculeInitial(System_Info%Total_Lyn_Count))
  ALLOCATE (SykMoleculeInitial(System_Info%Total_Syk_Count))
  ALLOCATE (RecMoleculePrevious(System_Info%Total_Rec_Count))
  ALLOCATE (LynMoleculePrevious(System_Info%Total_Lyn_Count))
```

```
 ALLOCATE (SykMoleculePrevious(System_Info%Total_Syk_Count))

 ! Read intial position of Receptor Molecule
 WRITE(*,*) '    read initial particle positions..'
 DO k = 1,System_Info%Total_Rec_Count

   READ(2,103) &
       RecMolecule(k)%Position(1),RecMolecule(k)%Position(2), &
       RecMolecule(k)%Position(3),RecMolecule(k)%Bond,
RecMolecule(k)%Agg_Size, &
       RecMolecule(k)%Boss, RecMolecule(k)%BoundRec_1,
RecMolecule(k)%BoundRec_2

   RecMolecule(k)%RecID = k ! restored 02-23-2017


   ! use modulo to shift all initial positions into the simulation box
   ! should work if (1) modulo is always non-negative (mod(5,3)=2 and mod(-1,3)=3)
   !              (2) xlimmax > xlimmin, same for the y bounds
   RecMolecule(k)%Position(1) = xlimmin + mod( RecMolecule(k)%Position(1),
xlimmax-xlimmin)
   RecMolecule(k)%Position(2) = ylimmin + mod( RecMolecule(k)%Position(2),
ylimmax-ylimmin)
   RecMolecule(k)%Position(3) = 0 !set z position to zero by hand



1103 FORMAT(F18.10,F18.10,F18.10,I2,I2,I6,I6,I6)

   if (System_Info%OutputLevel>=1) &
       write(*,&
       FMT="('RecID ',I3,' Coord ',3(' ',f10.6),' Bond Size Boss Buddies ',5(I3,' '))"),&
       k, RecMolecule(k)%Position,&
       RecMolecule(k)%Bond, RecMolecule(k)%Agg_Size,  RecMolecule(k)%Boss, &
       RecMolecule(k)%BoundRec_1, RecMolecule(k)%BoundRec_2



103  FORMAT(F18.16,F18.16,F18.16,I2,I2,I3,I2,I2)
 END DO

 CLOSE(2)

 ! Reading of lyn paramters
 WRITE (*,*) 'read lyn positions here'
 DO iLyn = 1,System_Info%Total_Lyn_Count
```

```
    READ (112,1104) &
        LynMolecule(iLyn)%Position(1), LynMolecule(iLyn)%Position(2), &
        LynMolecule(iLyn)%Position(3), LynMolecule(iLyn)%Receptor_ID
1104 FORMAT(F18.16,F18.16,F18.16,I2)

    ! use modulo to shift all initial positions into the simulation box
    ! should work if (1) modulo is always non-negative (mod(5,3)=2 and mod(-1,3)=3)
    !             (2) xlimmax > xlimmin, same for the y bounds
    LynMolecule(iLyn)%Position(1) = xlimmin + mod(LynMolecule(iLyn)%Position(1),
xlimmax-xlimmin)
    LynMolecule(iLyn)%Position(2) = ylimmin + mod(LynMolecule(iLyn)%Position(2),
ylimmax-ylimmin)
    LynMolecule(iLyn)%Position(3) = 0

    LynMolecule(iLyn)%Phos = 0

    LynMolecule(iLyn)%Itam_site = 0
    LynMolecule(iLyn)%Lyn_site = 0

    ! TODO -- really not much info here, this is just to avoid error messages
    if (LynMolecule(iLyn)%Receptor_ID > 0) then
      LynMolecule(iLyn)%Itam_site = 1
      LynMolecule(iLyn)%Lyn_site = 1
    endif


    if (System_Info%OutputLevel>=1) &
        write(*,FMT="('LynID ',I3,' Coord ',3(' ',f18.16),' Rec ',i2)"), iLyn,
LynMolecule(iLyn)%Position,&
        LynMolecule(iLyn)%Receptor_ID
  END DO
  CLOSE(112)

  WRITE (*,*) 'read syk positions here'
  DO iSyk = 1,System_Info%Total_Syk_Count
    READ (13,1105) &
        SykMolecule(iSyk)%Position(1), SykMolecule(iSyk)%Position(2), &
        SykMolecule(iSyk)%Position(3), SykMolecule(iSyk)%Receptor_ID

    ! use modulo to shift all initial positions into the simulation box
    ! should work if (1) modulo is always non-negative (mod(5,3)=2 and mod(-1,3)=3)
    !             (2) xlimmax > xlimmin, same for the y bounds
    SykMolecule(iSyk)%Position(1) = xlimmin + mod(SykMolecule(iSyk)%Position(1),
xlimmax-xlimmin)
    SykMolecule(iSyk)%Position(2) = ylimmin + mod(SykMolecule(iSyk)%Position(2),
ylimmax-ylimmin)
```

```
    SykMolecule(iSyk)%Position(3) = - mod(abs(SykMolecule(iSyk)%Position(3)),
abs(zlimmax-zlimmin))

    SykMolecule(iSyk)%Itam_site=0;
    SykMolecule(iSyk)%Syk_site=0;

    if (SykMolecule(iSyk)%Receptor_ID > 0) then
      ! TODO: this should be updated when the input files are
      SykMolecule(iSyk)%Itam_site=1;
      SykMolecule(iSyk)%Syk_site=1;
    endif

    SykMolecule(iSyk)%Phos = 0

1105 FORMAT(F18.16,F18.16,F19.16,I2)
    ! NOTE -- because the z coordinate is negative, you need 3 extra characters
    !        using f18.16 shifts the read and the final decimal is read as the
    !        next thing, i.e. the bound receptor

    if (System_Info%OutputLevel>=1) &
        write(*,FMT="('SykID ',I3,' Coord ',3(' ',f18.16),' Rec ',i2)"), iSyk,
SykMolecule(iSyk)%Position,&
        SykMolecule(iSyk)%Receptor_ID
  END DO
  CLOSE(13)



  ! TODO: either read the initial positions of Lyn and Syk from a file (eg. two new input
files)
  !       or generate random initial positions right here
  !       one way or another, positions of Lyn and Syk should be set up here

  ! Calculate Diffusion Standard Deviation for each species type
  DiffSTD = sqrt(2*DiffCoeff_Monomer*dt) ! only one type is wortth pre-calculating
  DiffSTD_Lyn = sqrt(2*Lyn_DiffCoeff_Monomer*dt)
  DiffSTD_Syk = sqrt(2*Syk_DiffCoeff_Monomer*dt)
  !DiffSTD(1) = sqrt(2*DiffCoeff_Monomer*dt) ! Activated receptor: R
  !DiffSTD(2) = sqrt(2*(DiffCoeff_Monomer/Agg_Size)*dt) ! Resting Receptor: RR

  ! Define Initial Positions and states
  ! .. as copies of the initial large particle structs
  RecMoleculeInitial = RecMolecule
  LynMoleculeInitial = LynMolecule
  SykMoleculeInitial = SykMolecule
  ! also the "Previous" set for comparing at periodic printouts
  RecMoleculePrevious = RecMolecule
  LynMoleculePrevious = LynMolecule
  SykMoleculePrevious = SykMolecule

  ! initial Lyn, Syk and Phos states
  ! TODO: This only works for the "everything off" initial condition
  RecMolecule%Iga_Phos = 0 ! 0= unphosphorylated, 1 = singly phosphorylated, 2 =
double phosphorylated
  RecMolecule%Iga_Lyn = 0! 0 no Lyn, some number is Lyn ID
  RecMolecule%Iga_Syk = 0 ! 0 no Syk, some number is Syk ID
  RecMolecule%Igb_Phos = 0 ! 0= unphosphorylated, 1 = singly phosphorylated, 2 =
double phosphorylated
  RecMolecule%Igb_Lyn = 0! ! 0 no Lyn, some number is Lyn ID
  RecMolecule%Igb_Syk = 0 ! 0 no Syk, some number is Syk ID

  RecMolecule%Domain = 0! assume free

  !Define initial boss
  RecMolecule%Boss = RecMolecule(:)%Boss ! ??

  !Initialize the last on-off time
  RecMolecule%LastOnOffTime = 0 !

  ! ***  identify the initial domain for each particle ***
  DomainParticleCount = 0 ! set the counte to zero for each domain

  do i=1,NP

    ParticleDomain=0

    do ii=1,NumDomains

      if(InDomain(RecMolecule(i)%Position(1:2),ii)) then
        ParticleDomain(ii)=1
        RecMolecule(i)%Domain = ii

      endif

    end do

    if (RecMolecule(i)%Domain>0) &
       DomainParticleCount(RecMolecule(i)%Domain) =
DomainParticleCount(RecMolecule(i)%Domain)+1
```

```
    ! formatted printout of initial receptor positions and containing domains
    write(*,*) 'RecID=',i,' Coord ',RecMolecule(i)%Position(1:2),' Dom ',
RecMolecule(i)%Domain, ' InDom',sum(ParticleDomain(1:5))
    !write(*,*) RecMolecule(i)%Position(1:2), RecMolecule(i)%Domain

  end do

  write(*,*) 'Total receptors: ', System_Info%Total_Rec_Count,' by domain: ',
DomainParticleCount(1:NumDomains),&
       ' untrapped: ', System_Info%Total_Rec_Count -
sum(DomainParticleCount(1:NumDomains))

  ! Initialize Time counter
  st=0 ! Time
  m=0 ! writing filename counter
  lifecount=0 ! lifetime update counter
  p=0 ! print counter
  ! Initialize dephosphorylation event counter
  !dephosevent=0

  ! Generate seed from system clock
  CALL SYSTEM_CLOCK(COUNT=seed)
  ! Seed grnd()  ! USE THIS FOR SIMULATIONS

  seed=1234 ! fixed seed ensures the same random numers each time, USE FOR
DEVELOPMENT ONLY
  if (System_Info%OutputLevel >=1) write(*,*) 'WARNING: Using fixed random
seed:',seed

  CALL sgrnd(seed)


  ! Calculate number of loop steps
  moves=N*NPT
  ! Calculate time step per particle
  pdt=dt/NPT

  ! Main simulation loop
  DO j=1,moves

    ! Update Time
    st=j*pdt
    System_Info%Current_Simulation_Time = st

    ! *** Diffusion & Kinetic Portion of Code ***
    ! Pick particle to move/react from among ALL particles,
    ! i.e. receptors, Lyn, Syk free or bound

    tt =
System_Info%Total_Rec_Count+System_Info%Total_Lyn_Count+System_Info%Total_
Syk_Count
    if (System_Info%OutputLevel >=2) write(*,*) 'Main loop pass.. Particle counts:', &
        'Total:',tt,&
        'Rec:',System_Info%Total_Rec_Count,&
        'Lyn:',System_Info%Total_Lyn_Count,&
        'Syk:',System_Info%Total_Syk_Count

    i = 1 + int(tt * real(grnd(), 16)) ! grnd() is simple "real", we convert it to double -- this
version is ok on a Mac
    !i = 1 + int(tt * real(grnd())) ! grnd() is simple "real", we convert it to double -- need
this for PC ?
    if (i>tt) i=tt

    if (System_Info%OutputLevel >=2) write(*,*) 'Chosen particle: ', i

    ! [re]set the reaction indicator to false
    System_Info%Reaction=.false.

    ! *** diffusion, binding, and unbinding (d/b/u) ***
    if (i <= System_Info%Total_Rec_Count) then ! this branch for receptors

      ! ** Receptor Branch (d/b/u) **
      if (System_Info%OutputLevel >=2) write(*,*) 'DBU - Receptor no.',i



      ! diffusion, binding, and unbinding are implemented by aggregate (bosses only)
      IF (RecMolecule(i)%Boss == i .AND. RecMolecule(i)%BoundRec_1 == 0) THEN

        ! diffusion also checks for binding (if it occurs, System_Info%Reaction will be set
to true)
        CALL ParticleDiffuse(i, DiffSTD / RecMolecule(i)%Agg_Size )

        ! unbinding only happens for bosses that have not undergone binding
        IF (RecMolecule(i)%Agg_Size > 1 .and. (System_Info%Reaction .eqv. .false.) )
THEN

          rannum = grnd()
          UnbindProb = Dimer_off_rate * (RecMolecule(i)%Agg_Size - 1) * dt

          IF (rannum <= UnbindProb) THEN ! undimerize
```

```
            CALL UnbindReaction(i)
            System_Info%Reaction=.true.
          END IF

        END IF

      END IF ! if this particle is a boss

      ! ** end of Receptor branch for diffusion, binding and unbinding (d/b/u) **

    else if  (i <= (System_Info%Total_Rec_Count + System_Info%Total_Lyn_Count)) then ! Lyn branch

      ! ** Lyn d/b/u/ branch **
      iLyn = i-System_Info%Total_Rec_Count

      if (System_Info%OutputLevel >=2) write(*,*) 'DBU - Lyn no.',iLyn

      IF (LynMolecule(iLyn)%Receptor_ID == 0) THEN
        ! if this Lyn is free, then it diffuses and may bind to a receptor
        CALL LynDiffuse (iLyn, DiffSTD_Lyn)
      ELSE
        ! if bound to a receptor, it may unbind
        rannum = grnd()

        ! The lyn off-rate depends on how it is bound to the receptor- through it's unique domain or SH2
        UnBindProb=Lyn_dimer_off_rate(LynMolecule(iLyn)%Lyn_site)*dt ! Lyn_site = 1 (unique domain) OR Lyn_site =2 (SH2 domain)

        IF (rannum <= UnbindProb) THEN ! unbind
          CALL LynUnbindReaction(iLyn)
          System_Info%Reaction=.true.
        END IF

      END IF

      ! ** end of Lyn d/b/u branch **

    else if  (i <= (System_Info%Total_Rec_Count + System_Info%Total_Lyn_Count + System_Info%Total_Syk_Count)) then ! Syk branch

      ! ** Syk d/b/u branch **

      iSyk = i-System_Info%Total_Rec_Count - System_Info%Total_Lyn_Count
      if (System_Info%Reaction .or. System_Info%OutputLevel >=2) write(*,*) 'DBU -
Syk no.',iSyk

      Syk_Pick_Count = Syk_Pick_Count + 1

      IF (SykMolecule(iSyk)%Receptor_ID == 0) THEN
        ! if this Syk is free, then it diffuses and may bind to a receptor
        Syk_FreePick = Syk_FreePick+1
        CALL SykDiffuse (iSyk, DiffSTD_Syk)

      ELSE
        ! if this Syk is bound to a receptor, it may unbind (diffusion triggered by the
aggregate..)
        ! TODO : check that probabilities are right
        Syk_BoundPick = Syk_BoundPick+1
        ! Syk dimer off rate will depend on SH2 binding site (1=Iga, 2=Igb)
        UnbindProb = Syk_dimer_off_rate(SykMolecule(iSyk)%Syk_site) * dt

        rannum = grnd()
        IF (rannum <= UnbindProb) THEN ! undimerize
          CALL SykUnbindReaction(iSyk)
          System_Info%Reaction=.true.
          Syk_BoundPick_Unbound = Syk_BoundPick_Unbound + 1
        END IF

      END IF

      if (System_Info%Reaction .eqv. .false.) Syk_PickNoReaction =
Syk_PickNoReaction + 1

      ! ** end of Syk d/b/u branch **
      ! ** Rec, Lyn, Syk d/b/u branches meet **

    end if

    ! done with diffusion, binding, unbinding
    ! may have resulted in aggregate changes (System_Info%Reaction)

    ! next, implement intrinsically triggered processes, OTHER THAN dissociation
    ! each process type is visited when the PARTICLE THAT IT AFFECTS is chosen

    if (System_Info%Reaction .eqv. .false.) then ! all intrinsic reactions are off if a
binding/unbinding reaction has occurred

      ! intrinsic reactions next ...
```

```
     if (i <= System_Info%Total_Rec_Count) then ! intrinsic branch for receptors only

       ! Principle: Intrinsic reactions of receptors are triggered via receptors - 05-18-2017
       ! ALL receptors picked will go through this branch (not only bosses)
       ! the only reactions we are concerned with here are receptor phos and dephos
       ! the rates depend on presence of and state of Lyn on an adjacent receptor
       ! and of the phos state of each site

       ! ** Compute probabilities for intrinsic reactions **
       ! they apply to all receptors, but are treated as mutually exclusive
       ! with binding and unbinding

       ! NOTE: Lyn and Syk binding are not intrinsic reactions (if Lyn and Syk are
spatial)

       ! * Phosphorylation  / Dephosphorylation probabilities *

       ! gather some info on the aggregate containing this receptor

       ! the probabilities below are vectors
       ! default is zero, will enter nonzero values as needed
       DePhosProb=0
       PhosProb = 0

       ! Receptor Phosphorylation (Lyn mediated)

       ! 05-18-2017 -- requires presence of Lyn on an immediately ADJACENT receptor
       !          number of Lyn's found does not matter, only whether
       !          (1) there is a Lyn (2) there is an activated Lyn

       ! * determine the presence and most active state of a Lyn on adjacent receptors *
       NeighborLynState=0; ! 0 means no Lyn present
       do iBR=1,2   ! loop over neighbor to  the left and right
         BoundRecID = RecMolecule(i)%BoundRec_1
         if (iBR==2) BoundRecID = RecMolecule(i)%BoundRec_2

         if (BoundRecID > 0) then ! only go on if there is a receptor there
           do iITAM=1,2 ! loop over ig alpha ig beta
             NeighborLynID=RecMolecule(BoundRecID)%Iga_Lyn
             if (iITAM==2)  NeighborLynID=RecMolecule(BoundRecID)%Igb_Lyn

             if (NeighborLynID>0) then
               NeighborLynState = max(NeighborLynState,1) ! a Lyn is present so raise
state to >=1; Inactive lyn
               ! check the state of the Lyn
```

```
                if (LynMolecule(NeighborLynID)%Phos >0)  NeighborLynState =2;
!Active Lyn
              endif

            enddo ! loop over itams
          endif ! if there is a receptor bound
        enddo ! loop over neighbors

        if (System_Info%OutputLevel >=3) &
           write(*,FMT="('RecID ',I3,' Intrinsic --  AggSize ',I3, '  Lyn status ', I2)"),&
           i,RecMolecule(i)%Agg_Size,NeighborLynState

        ! Phos probabilities depend on ITAM state and activity of neighboring Lyn, and
the state of the substrate ITAM
        ! Phos_rate[IA(0 phos), IA(1Phos), A(0 Phos), A(1 Phos)]; IA = Inactive Lyn, A=
Active Lyn
        if (NeighborLynState > 0) then ! Lyn is present (Inactive=1, Active=2)

          ! Deal with Igalpha
          if (RecMolecule(i)%Iga_Phos < 2) PhosProb(1) =
Phos_rate(2*NeighborLynState - 1 + RecMolecule(i)%Iga_Phos) * dt
          ! Deal with Igbeta
          if (RecMolecule(i)%Igb_Phos < 2) PhosProb(2) =
Phos_rate(2*NeighborLynState - 1 + RecMolecule(i)%Igb_Phos) * dt

          if (System_Info%OutputLevel >=3) then
            write(*,FMT="('A RecID ',I3,' Intrinsic --  AggSize ',I3, '  Lyn status ', I2,'
PhosProb=',2(e10.4,' '))"),&
               i,RecMolecule(i)%Agg_Size,NeighborLynState, PhosProb

          endif
        end if ! if NeighborLynState>0

        ! Dephosphorylation probs are the same for IgA,IgB; depend on phos state (2->1
or 1->0); docked Lyn or Syk protects
        ! Dephos probablity [1P, 2P]
        ! Deal with Igalpha
        if  (RecMolecule(i)%Iga_Phos >= 1 .AND. RecMolecule(i)%Iga_Lyn == 0 .AND.
RecMolecule(i)%Iga_Syk == 0) &! tyrosines are not protected
           DePhosProb(1) = Phos_off_rate(RecMolecule(i)%Iga_Phos) * dt
        ! Deal with Igbeta
        if  (RecMolecule(i)%Igb_Phos >= 1 .AND. RecMolecule(i)%Igb_Lyn == 0 .AND.
RecMolecule(i)%Igb_Syk == 0) &! tyrosines are not protected
           DePhosProb(2) = Phos_off_rate(RecMolecule(i)%Igb_Phos) * dt

        ! Put the phos/dephos probs into a vector
        ProbVec(1:2)=PhosProb !  (1 = Iga+, 2 = Igb+
        ProbVec(3:4)=DePhosProb !(3 = Iga-, 4 = Igb-)
        ProbVec(5)=1-sum(ProbVec(1:4)) ! for the null event included explicitely to have
a normalized probability vector

        if (NeighborLynState > 0 .and. System_Info%OutputLevel >=3) then

          write(*,FMT="('C RecID ',I3, '  Lyn status ', I2,' PhosProb=[',2(e12.4,' '),']')"),&
             i,NeighborLynState, PhosProb
          write(*,FMT="('  cont  ProbVec = [ ',4(e12.4,' '),']')"), ProbVec


        endif

        ! TODOx -- only do the following if the probabilities are not all zero
        ! choose exactly one outcome (including non-event)

        if(sum(ProbVec)>0) then

          rannum = grnd()

          ! unfortunately this is necessary
          do while(rannum > 1.0 .and. rannum < 0.0)
            rannum = grnd()
          enddo

          SumProb=0
          PhosIndex=0 !
          do while(rannum > SumProb)

            PhosIndex=PhosIndex+1
            SumProb = SumProb + ProbVec(PhosIndex)

          enddo
          ! TODO - check behavior when rannum=0 or 1
          ! TODOx perhaps do this for safety --
          if(rannum<=0) PhosIndex=1
          ! if(rannum>=1) PhosIndex=4

          if (NeighborLynState > 0 .and.  System_Info%OutputLevel >=3) then
            write(*,FMT="('RecID ',I3,' NLS=',I2,' PV=[',5(e10.4,' '),'] Ind=',I1,'
SP=',e10.4,' ran=',e10.4)"),&
               i,NeighborLynState,ProbVec,PhosIndex,SumProb,rannum
          endif

          ! p has the chosen event type
          if(PhosIndex<5) then

            if (System_Info%OutputLevel >=3) &
               write(*,FMT="('Phos Choice -- PV:',8(f8.6,'  '),' r=',f8.6,' p=',i0,' CP=',e10.6
)") &
               ProbVec, rannum, PhosIndex, SumProb

            System_Info%Reaction = .true.

            if     (PhosIndex == 1) then ! Iga Phos
              RecMolecule(i)%Iga_Phos =  RecMolecule(i)%Iga_Phos + 1
              ! if Iga reached Ph=2, update the binding mode on any bound Syk
              if(RecMolecule(i)%Iga_Phos==2 .AND. RecMolecule(i)%Iga_Syk>0 ) &
                  SykMolecule( RecMolecule(i)%Iga_Syk )%Syk_site = 2
            else if (PhosIndex == 2) then ! Igb Phos
              RecMolecule(i)%Igb_Phos =  RecMolecule(i)%Igb_Phos + 1
              ! if Igb reached Ph=2, update the binding mode on any bound Syk
              if(RecMolecule(i)%Igb_Phos==2 .AND. RecMolecule(i)%Igb_Syk>0 ) &
                  SykMolecule( RecMolecule(i)%Igb_Syk )%Syk_site = 2
            else if (PhosIndex == 3) then ! Iga Dephos
              RecMolecule(i)%Iga_Phos =  RecMolecule(i)%Iga_Phos - 1
            else if (PhosIndex == 4) then ! Igb Dephos
              RecMolecule(i)%Igb_Phos =  RecMolecule(i)%Igb_Phos - 1
            end if



            ! log output
            if (System_Info%OutputLevel >=1) then
              if (PhosIndex<=2) then ! Phos
                write(*,FMT="('T=',f14.8,' RecPhos RecID=',i3,' Ig:',i1,'(',(2i1),')
Agg=',i3,' Sz=',i0)") &

System_Info%Current_Simulation_Time,i,PhosIndex,RecMolecule(i)%Iga_Phos,RecMo
lecule(i)%Igb_Phos,&
                 RecMolecule(i)%Boss,RecMolecule(i)%Agg_Size
              else  ! Dephos
                write(*,FMT="('T=',f14.8,' RecDeph RecID=',i3,' Ig:',i1,'(',(2i1),')
Agg=',i3,' Sz=',i0)") &
                 System_Info%Current_Simulation_Time,i,PhosIndex-
2,RecMolecule(i)%Iga_Phos,RecMolecule(i)%Igb_Phos,&
                 RecMolecule(i)%Boss,RecMolecule(i)%Agg_Size
              endif
            endif

          else ! non-event
```

```
          endif ! if PhosIndex < 5

        endif ! if sum(ProbVec)>0

        ! * end intrinsic reactions Rec branch * !

      else if  (i <= (System_Info%Total_Rec_Count + System_Info%Total_Lyn_Count))
then ! Lyn branch

        ! ** Lyn phos/dephos branch **
        iLyn = i-System_Info%Total_Rec_Count!  ID of Lyn particle

        IF (LynMolecule(iLyn)%Phos == 0) then !  phos site on lyn is 0 --> check for
phos

          ! EXPLANATION: Lyn phos must be mediated by another Lyn
          !            substrate and activator Lyn's must be *bound to adjacent receptors*
          !            the substrate Lyn is the one we are updating

          if (LynMolecule(iLyn)%Itam_site> 0) then ! require lyn bound to a receptor

            ! phos possible only if nearby lyns are present

            ! id of receptor this Lyn is bound to
            LynRecID = LynMolecule(iLyn)%Receptor_ID

            ! * determine the presence and most active state of a Lyn on adjacent receptors
*
            NeighborLynState=0; ! 0 means no Lyn present
            do iBR=1,2   ! loop over neighbor to  the left and right
              BoundRecID = RecMolecule(LynRecID)%BoundRec_1
              if (iBR==2) BoundRecID = RecMolecule(LynRecID)%BoundRec_2

              if (BoundRecID > 0) then ! only go on if there is a receptor there
                do iITAM=1,2 ! loop over ig alpha ig beta
                  NeighborLynID=RecMolecule(BoundRecID)%Iga_Lyn
                  if (iITAM==2)  NeighborLynID=RecMolecule(BoundRecID)%Igb_Lyn

                  if (NeighborLynID>0) then
                    NeighborLynState = max(NeighborLynState,1) ! a Lyn is present so
raise state to >=1; Inactive lyn
                    ! check the state of the Lyn
                    if (LynMolecule(NeighborLynID)%Phos >0)  NeighborLynState =2;
!Active Lyn
                  endif
                enddo ! loop over itams
              endif ! if there is a receptor bound
            enddo ! loop over neighbors


            if (NeighborLynState > 0) then ! if Lyn is present..


              rannum = grnd() ! implement phos with appropriate rate

              if (System_Info%OutputLevel >=3) &
                 write(*,FMT="('LynID ',I3,' RecID ',I3,' NLS=',I2,' Pr=',e10.4,'
ran=',e10.4)"),&
                 iLyn,LynRecID,NeighborLynState,Lyn_Phos_Rate(NeighborLynState) *
dt,rannum

              if (rannum <= Lyn_Phos_Rate(NeighborLynState) * dt) then

                LynMolecule(iLyn)%Phos=1 ! phosphorylation

                System_Info%Reaction = .true.

                if (System_Info%OutputLevel >=1) & ! log output for Lyn
phosphorylation
                   write(*,FMT="('T=',f14.8,' LynPhos LynID=',i3,' RecID=',i3,' Agg=',i3,'
Sz=',i0)") &
                   System_Info%Current_Simulation_Time,iLyn,LynRecID,&
                   RecMolecule(LynRecID)%Boss,RecMolecule(LynRecID)%Agg_Size

              endif ! if phos happens
            endif ! if activator present

          endif ! if this Lyn is receptor bound

        ELSE !phos site on lyn is >0 -- dephos possible

          rannum = grnd()

          if (rannum <= Lyn_Dephos_Rate * dt ) then

            LynMolecule(iLyn)%Phos = 0 ! lyn dephosphorylation

            System_Info%Reaction = .true.

            if (System_Info%OutputLevel >=1) & ! log output for Lyn de-phosphorylation
```

```
                write(*,FMT="('T=',f14.8,' LynDeph LynID=',i3,' RecID=',i3,' Agg=',i3,'
Sz=',i0)") &
                System_Info%Current_Simulation_Time,iLyn,LynRecID,&
                RecMolecule(LynRecID)%Boss,RecMolecule(LynRecID)%Agg_Size
          endif

        END if ! if Lyn phos state=0

        ! * end intrinsic reactions Lyn branch *

      else if  (i <= (System_Info%Total_Rec_Count + System_Info%Total_Lyn_Count +
System_Info%Total_Syk_Count)) then ! Syk branch

        ! ** intrinsic reactions (phos/dephos) Syk branch **
        Syk_Intrinsic_Count = Syk_Intrinsic_Count + 1

        ! ID of the Syk being updated
        iSyk = i-System_Info%Total_Rec_Count - System_Info%Total_Lyn_Count

        ! SykPhos reactions..

        !IF (SykMolecule(iSyk)%Phos == 0 .OR. SykMolecule(iSyk)%Phos_2 == 0 ) then
!  phos site on syk is 0 --> check for phos

        ! EXPLANATION: Syk phos possible only if
        ! (1) nearby syks are present (on adjacent receptors, similar to Lyn) or
        ! (2) syk_site = 2 (tandem SH2 domains are engaged with an ITAM)
        ! both require substrate docked on receptor
        ! boh lead to the same outcome, so probs will be added up

        if (SykMolecule(iSyk)%Itam_site > 0) then ! is syk bound to a receptor ?
          ! id of receptor this Syk is bound to
          SykRecID = SykMolecule(iSyk)%Receptor_ID
        else
          ! Syk is free -- may still dephosphorylate
          SykRecID = 0
        endif



        ! Syk has two phosphorylation sites: one is phosphorylated by Syk (phos) and the
other by Lyn (phos_2)
        !    there is a dephos. rate for each site
        !
        ! Calculate the probability of each [de]phosphorylation
        ! (1) If site 1 is active --> DePhos1
        !            inactive --> Phos1; may happen only if NeighborSykState > 0
        !
        ! (2) If site 2 is active --> DePhos2
        !           inactive  --> Phos 2; may happen only if NeighborLynState > 0
        !
        ! NOTE: we avoid implementing two transformations in the same update, so we
        !      assign / split the (very small) prob. of both sites changing states back to
        !      the prob. of one site changing

        ! re-use PhosProb(2), DePhosProb(2), ProbVec(5)
        PhosProb=0
        DePhosProb=0
        ProbVec=0



        if (SykMolecule(iSyk)%Phos == 1) then
          ! dephos - prob > 0 only if the site is active
          DePhosProb(1) = Syk_DePhos_Rate * dt ! approximates 1 - exp( -
Syk_DePhos_Rate * dt )
        else
          ! phos 1 - site must be inactive and there must be an adjacent Syk
          ! * determine the presence and most active state of a Syk on adjacent receptors *
          NeighborSykState=0; ! 0 means no adjacent Syk present
          if (SykRecID > 0) then
            do iBR=1,2   ! loop over neighbor to  the left and right
              BoundRecID = RecMolecule(SykRecID)%BoundRec_1
              if (iBR==2) BoundRecID = RecMolecule(SykRecID)%BoundRec_2

              if (BoundRecID > 0) then ! only go on if there is a receptor there
                do iITAM=1,2 ! loop over ig alpha ig beta
                  NeighborLynID=RecMolecule(BoundRecID)%Iga_Syk
                  if (iITAM==2)  NeighborLynID=RecMolecule(BoundRecID)%Igb_Syk

                  if (NeighborSykID>0) then
                    NeighborSykState = max(NeighborSykState,1) ! a Syk is present so
raise state to >=1; Inactive Syk
                    ! check the state of the Syk
                    if (SykMolecule(NeighborSykID)%Phos >0)  NeighborSykState =2;
!Active Syk
                  endif

                enddo ! loop over itams
              endif ! if there is a receptor bound
            enddo ! loop over neighbors
            if (NeighborSykState>0) PhosProb(1) = Syk_Phos_Rate(NeighborSykState) *
dt
            ! end Syk phos site 1 inactive case
          endif ! if SykRecID > 0
        endif ! if Syk phos site 1 is active ..

        if (SykMolecule(iSyk)%Phos_2 == 1) then
          ! dephos - prob > 0 only if the site is active
          DePhosProb(2) = Syk_DePhos_Rate * dt
        else
          ! phos 2 - site must be inactive and there must be an adjacent Lyn
          ! * determine the presence and most active state of a Lyn on adjacent receptors *
          ! LynRecID = LynMolecule(iLyn)%Receptor_ID! TODO change this to
SykRecID
          NeighborLynState=0; ! 0 means no Lyn present
          if (SykRecID > 0 ) then
            do iBR=1,2   ! loop over neighbor to  the left and right
              BoundRecID = RecMolecule(SykRecID)%BoundRec_1
              if (iBR==2) BoundRecID = RecMolecule(SykRecID)%BoundRec_2

              if (BoundRecID > 0) then ! only go on if there is a receptor there
                do iITAM=1,2 ! loop over ig alpha ig beta
                  NeighborLynID=RecMolecule(BoundRecID)%Iga_Lyn
                  if (iITAM==2)  NeighborLynID=RecMolecule(BoundRecID)%Igb_Lyn

                  if (NeighborLynID>0) then
                    NeighborLynState = max(NeighborLynState,1) ! a Lyn is present so
raise state to >=1; Inactive lyn
                    ! check the state of the Lyn
                    if (LynMolecule(NeighborLynID)%Phos >0)  NeighborLynState =2;
!Active Lyn
                  endif

                enddo ! loop over itams
              endif ! if there is a receptor bound
            enddo ! loop over neighbors
            if (NeighborLynState>0) PhosProb(2) = Lyn_Phos_Rate(NeighborLynState) *
dt
          endif ! if SykRecID>0
          ! end Syk phos site 2 inactive case
        endif ! if Syk phos site 2 is inactive ...

        ! Put the phos/dephos probs into a vector
        ProbVec(1:2)=PhosProb !  (1 = Phos1, 2=Phos2
        ProbVec(3:4)=DePhosProb !(3 = DePhos1, 4 = DePhos2)
        ProbVec(5)=1-sum(ProbVec(1:4)) ! for the null event included explicitely to have
a normalized probability vector
```

```
        if(sum(ProbVec(1:4))>0) then ! only go through the random selection if there is at
least one nozero prob 1:4

          ! Choose the outcome using a random number
          rannum = grnd()
          ! hopefully this is not necessary this time
          !do while(rannum > 1.0 .and. rannum < 0.0)
          !   rannum = grnd()
          !enddo

          SumProb=0
          PhosIndex=0 ! set to zero by default - note that the null event is 5
          do while(rannum > SumProb .and. PhosIndex < 5)

            PhosIndex=PhosIndex+1
            SumProb = SumProb + ProbVec(PhosIndex)

          enddo

          ! TODO - check behavior when rannum=0 or 1
          ! do this for safety -- strange ran() leads to null event
          if(rannum<=0 .or. rannum>=1) PhosIndex=5  ! 5 is the null event here

          ! endif

          !*****

          if (PhosIndex==1) then
            SykMolecule(iSyk)%Phos = SykMolecule(iSyk)%Phos + 1 ! phos and update
          elseif (PhosIndex==2) then
            SykMolecule(iSyk)%Phos_2 = SykMolecule(iSyk)%Phos_2 + 1 ! phos and
update
          elseif (PhosIndex==3) then
            SykMolecule(iSyk)%Phos = SykMolecule(iSyk)%Phos - 1 ! dephos and
update
          elseif (PhosIndex==4) then
            SykMolecule(iSyk)%Phos_2 = SykMolecule(iSyk)%Phos_2 - 1 ! dephos and
update
          endif

          System_Info%Reaction = .true.


            if (System_Info%OutputLevel >=1) then   !phosindex 1 = phos1, !phosindex 2
= phos2,
              if (PhosIndex<=2) then ! Phos
```

```
write(*,FMT="('T=',f14.8,' SykPhos SykID=',i3,' RecID=',i3,' Phos',i1,' Phos1:',i1,'
Phos2:',i1,' Agg=',i3,' Sz=',i0)") &

System_Info%Current_Simulation_Time,iSyk,SykRecID,PhosIndex,SykMolecule(iSyk)
%Phos, &
                    SykMolecule(iSyk)%Phos_2,
RecMolecule(SykRecID)%Boss,RecMolecule(SykRecID)%Agg_Size
           else if (PhosIndex<=4) then! Dephos
write(*,FMT="('T=',f14.8,' SykDeph SykID=',i3,' RecID=',i3,' Phos',i1,' Phos1:',i1,'
Phos2:',i1,' Agg=',i3,' Sz=',i0)") &
                    System_Info%Current_Simulation_Time,iSyk,SykRecID,PhosIndex-
2,SykMolecule(iSyk)%Phos, &

SykMolecule(iSyk)%Phos_2,RecMolecule(SykRecID)%Boss,RecMolecule(SykRecID)
%Agg_Size
              endif
            endif


          !endif ! if Syk is bound to an ITAM

        endif ! if sum(Prob(1:4) > 0 )

        ! * end intrinsic reactions Syk branch *

      end if ! i <= receptor count

    end if ! System_Info%Reaction .eqv. .false. (i.e. if no reaction happened)

    ! ** done with intrinsic reactions for Rec,Lyn,Syk **

    !  write(*,*) '.. done with intrinsic reactions for all'

    ! *** Record Keeping Portion of Code ***

    p=p+1 ! counts the iterations from the last frame export

    ! Check if the data should be written to a file
    IF (p == datacut) THEN

!       write(*,*) 'T=',System_Info%Current_Simulation_Time,&
!           'Syk Picks:',Syk_Pick_Count, ' P-NoRxn:',Syk_PickNoReaction, &
!           ' Bind:', Syk_BindCall_Count, ' B-Rxn:', Syk_Bind_Reaction, ' B-NoRxn:',
Syk_Bind_NoReaction,&
!           ' NoRxn-AvBossCt:',Enc_BossCount,float(Enc_BossCount)/float(
Syk_Bind_NoReaction),&
```

```
!          ' NoRxn-AvEligAggCt:',Enc_EligAggCount,float(Enc_EligAggCount)/float(Syk_Bind_NoReaction),&
!          ' NoRxn-SysPhosSiteCt:',Enc_SysPhosCount,float(Enc_SysPhosCount)/ float(Syk_Bind_NoReaction),&
!          ' MinDist:',sqrt(SysMinDist), sqrt(SysMindist) / Syk_BindRad_Dimer
    !    '  --free:' , Syk_FreePick,  ' Intrinsic:',Syk_Intrinsic_Count
    !  ' -- bound:', Syk_BoundPick, ' --bound-unbound:',Syk_BoundPick_Unbound, &

    ! set all the above counters to zero
    Syk_Pick_Count=0
    Syk_Intrinsic_Count=0
    Syk_PickNoReaction=0
    Syk_FreePick=0
    Syk_BoundPick=0
    Syk_BoundPick_Unbound=0
    Syk_DiffCall_Count=0
    Syk_Diff_Reaction=0
    Syk_Diff_NoReaction=0
    Syk_BindCall_Count=0
    Syk_Bind_Reaction=0
    Syk_Bind_NoReaction=0
    Enc_EligAggCount=0
    Enc_BossCount=0
    Enc_SysPhosCount=0
    ! min distance - set to very large to begin the next pass
    SysMinDist=100

    if (System_Info%OutputLevel >=2) write(*,*) ' Printout ',m+1,'..'

    DO ii=1,NP
      RecMolecule(ii)%r_Squared=&
          (RecMolecule(ii)%Position(1)-RecMoleculePrevious(ii)%Position(1))**2 + &
          (RecMolecule(ii)%Position(2)-RecMoleculePrevious(ii)%Position(2))**2
    END DO

    ! Calculate OVERALL MSD (MSDx and MSDy too?) for specific dt
    MSD=SUM(RecMolecule(:)%r_Squared)/NP

    ! Keep the current state for comparison during the next printout
    RecMoleculePrevious = RecMolecule
    LynMoleculePrevious = LynMolecule
    SykMoleculePrevious = SykMolecule

    ! Restart dt counter
    m=m+1
```

```
     ! generate filename for data storage
     write(fnstring, fmt="(i0)") m
     open(77, file=trim(outdir)//'/Data_Files/ParticleData.'//trim(fnstring))
     WRITE(77,*) 'Rows: Particle Columns: Particle #, x, y, dx, dy, d, B1, &
        & B2, Bond, Agg_size, Boss'
     !& PhosEvent, # dephosphorylation events, RR Dimer Attempts, RR Dimmer
Successes'
     !WRITE(88,*) 'r^2 (Combined x and y moves), species'
     ! Write data to file
     DO i = 1,NP
       write(77,fmt="(&
          'RecID=',i4,' XYZ=[',3(f7.4,' '),'] BoundRec:',i3,' ',i3,' BondCt:',i1,'
AggBoss:',i4,&
          ' AggSz:',i2,' LynBound:',2(I3,' '),' SykBound:',2(I3,' '),' Ph:',i1,' ',i1,'
Dom:',i2)")&
          i,RecMolecule(i)%Position,&

RecMolecule(i)%BoundRec_1,RecMolecule(i)%BoundRec_2,RecMolecule(i)%Bond,&
          RecMolecule(i)%Boss, RecMolecule(i)%Agg_Size,&
          RecMolecule(i)%Iga_Lyn, RecMolecule(i)%Igb_Lyn,
RecMolecule(i)%Iga_Syk, RecMolecule(i)%Igb_Syk, &
          RecMolecule(i)%Iga_Phos, RecMolecule(i)%Igb_Phos,  &
          RecMolecule(i)%Domain

     END DO

     CLOSE(77)


     ! also write out Lyn and Syk
     ! -- in separate files (..?)
     ! * Lyn output *
     open(77, file=trim(outdir)//'/Data_Files/ParticleData_Lyn.'//trim(fnstring))
     write(77,*) 'Lyn section: LynID, x, y, z, RecID (if bound), Phos, ITAM site
(Iga/Igb), Lyn site (none/UD/SH2)'
     do i=1, Lyn_num
       write(77,fmt="('LynID=',i4,' XYZ=[',f8.4,' ',f8.4,' ',f8.4,'] Rec=',i4,' Ph=',i1,' Site
Ig:',i1,' Lyn:',i1)")&

i,LynMolecule(i)%Position,LynMolecule(i)%Receptor_ID,LynMolecule(i)%Phos,&
          LynMolecule(i)%Itam_site,LynMolecule(i)%Lyn_site
     end do
     close(77)

     ! * Syk output *
     open(77, file=trim(outdir)//'/Data_Files/ParticleData_Syk.'//trim(fnstring))
     write(77,*) 'Syk section: SykID, x, y, z, RecID (if bound), Phos, ITAM site
(Iga/Igb), Syk site (none/one SH2/two SH2)'
     do i=1, Syk_num
       write(77,fmt="('SykID=',i4,' XYZ=[',f8.4,' ',f8.4,' ',f8.4,'] Rec=',i4,' Ph=',i1,' Site
Ig:',i1,' Syk:',i1)")&

i,SykMolecule(i)%Position,SykMolecule(i)%Receptor_ID,SykMolecule(i)%Phos,&
          SykMolecule(i)%Itam_site,SykMolecule(i)%Syk_site
     end do
     close(77)




     ! dimer lifetimes from the actual simulation, not the frame rate
     open(11,file=trim(outdir)//'/AggSizeCounts')

     WRITE(11,*)
System_Info%Current_Simulation_Time,System_Info%Num_Aggregates,
System_Info%AggSizeCount

     WRITE(7,*) MSD ! This is the mean square displacement from frame to frame
     p=0

     if (System_Info%OutputLevel >=2) write(*,*) '  done with printouts'

   END IF

   !    END SELECT
   ! write(*,*) 'Loop'

 END DO
 CLOSE(4)
 CLOSE(7)
 CLOSE(8)
 CLOSE(9)
 CLOSE(10)
 CLOSE(11)

 WRITE(*,*) 'FORTRAN Simulation Ended'

END PROGRAM Pre_BCR
```

```
!************************************************************************
*************************************************************************
**********

SUBROUTINE LynDiffuse (iLyn, MyDiffSTD)
  !
  !
  USE ParticleInfo
  USE ModelConstants
  USE mtmod
  !
  !
  IMPLICIT NONE
  !
  !! Declare variables
  INTEGER, INTENT (IN) :: iLyn ! chosen particle
  DOUBLE PRECISION, INTENT (IN) :: MyDiffSTD !   diffusion standard deviation
  DOUBLE PRECISION :: r1, r2, w1, w2, x1, y1,z1, rannum
  integer :: CanReact=0
  !
  !
  !! randomly make a trajectory for particles using mtmod.f90 for random numbers !!!
  !! Generate random number & Normally distribute random number !
http://www.taygeta.com/random/gaussian.html
  !!* Generate x move
  r1=2*grnd()-1
  r2=2*grnd()-1
  !! Check unit circle, if not in reject and try again
  w1=r1*r1+r2*r2
  DO WHILE (w1 > 1)
    !! Generate random number again
    r1=2*grnd()-1
    r2=2*grnd()-1
    !! Unit circle check
    w1=r1*r1+r2*r2
  END DO
  w2=sqrt((-2*log(w1))/w1)
  !! Normally distributed random # for distance
  x1=r1*w2
  !
  !! Generate y move
  r1=2*grnd()-1
  r2=2*grnd()-1
  !! Check unit circle, if not in reject and try again
  w1=r1*r1+r2*r2
  DO WHILE (w1 > 1)
```

```
    !! Generate random number again
    r1=2*grnd()-1
    r2=2*grnd()-1
    !! Unit circle check
    w1=r1*r1+r2*r2
  END DO
  w2=sqrt((-2*log(w1))/w1)
  y1=r2*w2
  !
  !! Account for diffusion coefficient based on species type
  y1=y1*MyDiffSTD
  x1=x1*MyDiffSTD
  z1=0      ! Lyn does not move in the z direction



  ! call periodic boundary condition
!!$  CALL
PeriodicBC(LynMolecule(iLyn)%Position(1)+x1,LynMolecule(iLyn)%Position(2)+y1,
&
!!$       LynMolecule(iLyn)%Position(3)+z1, LynMolecule(iLyn)%Position(1), &
!!$       LynMolecule(iLyn)%Position(2),LynMolecule(iLyn)%Position(3))

  CALL PeriodicBC2(&
      LynMolecule(iLyn)%Position(1)+x1,LynMolecule(iLyn)%Position(2)+y1, &
      LynMolecule(iLyn)%Position(1), LynMolecule(iLyn)%Position(2))

  LynMolecule(iLyn)%Position(3)=0

  ! activated Lyn is always able to bind
  CanReact = LynMolecule(iLyn)%Phos
  ! un-activated Lyn reacts with a probability
  if ( LynMolecule(iLyn)%Phos==0) then
    rannum=grnd()
    if (rannum <= Lyn_available_fraction) CanReact=1
  endif

  if(CanReact==1) CALL LynBindReaction(iLyn)

END SUBROUTINE LynDiffuse

!***********************************************************************
************************************************************************
*****************

SUBROUTINE LynBindReaction(iLyn)
```

```
  USE ParticleInfo
  USE ModelConstants
  USE mtmod

  IMPLICIT NONE

  INTEGER, INTENT(IN) ::  iLyn ! specific particle
  INTEGER :: receptor_ID, num, add_lyn, Free_Lyn_Site_Count, LuckyRec,
LuckyRecIndex
  DOUBLE PRECISION :: distsq, rannum
  ! arrays to hold aggregate info Lyn_Agg is actually a list of receptors
  INTEGER,DIMENSION(MaxAgg) :: Lyn_Agg=0 ! list of receptors in a given
aggregate

  real, dimension( 5, MaxAgg  ) :: array_1=0 ! 5 x NumParticles - probabilities of binding
or similar
  real, dimension( 5 * MaxAgg ) :: array_2=0, CumSum=0 ! same as above but vector
  integer :: d, CurrentParticle, ip, k, h, j, u,m_lyn,rv,cv, i
  integer :: AggRecCount ! number of receptors on the current aggregate

  if (System_Info%OutputLevel >=3) write(*,*) 'LynBind begin -- LynID ',iLyn

  System_Info%Reaction = .FALSE.

  ! Check for reaction
  do k = 1,System_Info%Num_Particles ! loop over particles (receptor monomers)

    ! Check if one of available receptors is close enough to react

    if (RecMolecule(k)%Boss == k) then

      !  need to know how many free Lyn binding domains on this aggregate

      AggRecCount = RecMolecule(k)%Agg_Size  ! number of receptors on this
aggregate
      !AggRecCount = count( RecMolecule(:)%Boss==k ) ! this should match %AggSize

      Lyn_Agg = pack(RecMolecule(:)%RecID, RecMolecule(:)%Boss==k)! holding the
recs in an aggregate

      !count each type of binding- each receptor- whether general lyn binding is possible
through Iga, Igb
      if (System_Info%OutputLevel >=3) then
        WRITE(*,*) ' Lyn_Agg has size ',size(Lyn_Agg),' AggRecCount=',AggRecCount
        do j=1,size(Lyn_Agg)
```

```
          write(*,*) 'entry',j,':',Lyn_Agg(j)
        end do
      endif

      !count each type of binding- each receptor- whether general lyn binding is possible
through Iga, Igb

      array_1 = 0! sets all the elements to 0

      Do j = 1,AggRecCount ! this follows the receptor
        ! TODO: merge branches (1,2,3) and (4,5) in a()
        ! This block will check for the Igalpha branch
        if (RecMolecule(Lyn_Agg(j))%Iga_Lyn == 0 .and.
RecMolecule(Lyn_Agg(j))%Iga_Syk == 0) then ! This will check if Iga has any Lyn at
all
          if (RecMolecule(Lyn_Agg(j))%Iga_Phos == 0) then
            array_1(1,j)= LynBindScaleFactor(1)! unique domain on lyn binds to site 1 on
receptors
          else if (RecMolecule(Lyn_Agg(j))%Iga_Phos == 1) then !
            array_1(2,j)=  LynBindScaleFactor(2)
          else
            array_1(3,j)=  LynBindScaleFactor(2)!(RecMolecule(Lyn_Agg(j))%Phos ==
2)the probability of binding to a twice phosphorylated itams sould be twice as much
          end if
        end if
        ! This block will check for Igbeta branch
        if (RecMolecule(Lyn_Agg(j))%Igb_Lyn == 0 .and.
RecMolecule(Lyn_Agg(j))%Igb_Syk == 0) then !This will check if Igb has any Lyn at all
          if (RecMolecule(Lyn_Agg(j))%Igb_Phos == 1) then
            array_1(4,j)= LynBindScaleFactor(3)! unique domain on lyn binds to site 1 on
receptors
          else if(RecMolecule(Lyn_Agg(j))%Igb_Phos == 2)  then
            array_1(5,j)=  LynBindScaleFactor(3) !if (RecMolecule(Lyn_Agg(j))%Phos
== 2)
          end if
        end if

      End Do

      ! Decide
      distsq=&  ! Calculate the distance between lyn and receptor
         (RecMolecule(k)%Position(1)-LynMolecule(iLyn)%Position(1))**2+&
         (RecMolecule(k)%Position(2)-LynMolecule(iLyn)%Position(2))**2

      if (distsq <= SUM(array_1(:,1:AggRecCount))*Lyn_BindRad_Dimer**2) THEN
! Lyn is within binding radius
```

```
        ! implement the binding ..
        array_1 = array_1/SUM(array_1(:,1:AggRecCount))
        array_2(1:5*AggRecCount) = reshape(array_1(:,1:AggRecCount),(/5*AggRecCount/)); !array 3 hold them in one dimensional vector

        CumSum(1)= array_2(1)
        Do i= 2,5*AggRecCount
          CumSum(i)= array_2(i)+CumSum(i-1)
        End do


        rannum = grnd()
        m_lyn = 0
        Do u = 1,5*AggRecCount
          if (rannum <= CumSum(u)) then
            m_lyn = u ! element in array 2
            exit
          end if
        End do


        cv= ceiling(REAL(m_lyn)/5)! this will give receptor number

        rv= mod(m_lyn,5) ! this will give row number
        if (rv == 0) rv=5! this is to make sure that row vector is never 0

        ! ** updates for successful binding go here **

        ! update the Lyn and the receptor

        LynMolecule(iLyn)%Receptor_ID = Lyn_Agg(cv)
        LynMolecule(iLyn)%Position = RecMolecule(Lyn_Agg(cv))%Position ! may not need this - set Position to zero instead ?

        ! identify the ITAM and binding mode
        If (rv <= 3) then ! Iga ..
          RecMolecule(Lyn_Agg(cv))%Iga_Lyn = iLyn
          LynMolecule(iLyn)%Itam_site = 1        ! Lyn is bound to Igalpha
          If (rv == 1) then
            LynMolecule(iLyn)%Lyn_site  = 1        ! Lyn is bound through it's unique domain
          else
            LynMolecule(iLyn)%Lyn_site  = 2
          end if                                ! Lyn is bound through it's SH2 domain
          else if (rv >= 4) then ! Igb
            RecMolecule(Lyn_Agg(cv))%Igb_Lyn = iLyn
            LynMolecule(iLyn)%Itam_site = 2        ! Lyn is bound to Igbeta
            LynMolecule(iLyn)%Lyn_site  = 2        ! Lyn is ALWAYS bound to Igbeta
through it's SH2 domain
          end if

          ! log output
          if (System_Info%OutputLevel >=1) &
              write(*,FMT="('T=',f14.8,' LynBind LynID=',i3,'.',i1,' RecID=',i3,'.',i1,'
Agg=',i3,' Sz=',i0,' Free Lyn ',i0)") &
              System_Info%Current_Simulation_Time,iLyn,LynMolecule(iLyn)%Lyn_site,
Lyn_Agg(cv),LynMolecule(iLyn)%Itam_site,&
              RecMolecule(Lyn_Agg(cv))%Boss,&
              RecMolecule(Lyn_Agg(cv))%Agg_Size,System_Info%Free_Lyn_Count

          ! free Lyn count
          System_Info%Free_Lyn_Count = System_Info%Free_Lyn_Count - 1

          ! flag to end the update pass
          System_Info%Reaction = .true.

        end if ! if receptor is within BR

    end if ! only if there are free lyn binding domains on the aggregate

    if (System_Info%Reaction) exit ! break the loop / so only one binding reaction

  end do ! loop over all receptors

  if (System_Info%OutputLevel >=3) write(*,*) 'LynBind end'


END SUBROUTINE LynBindReaction
!************************************************************************
*************************************************************************
*****************


SUBROUTINE LynUnbindReaction(iLyn)

  USE ParticleInfo
  USE ModelConstants
  USE mtmod

  IMPLICIT NONE
```

```
 INTEGER, INTENT (IN) :: iLyn
 DOUBLE PRECISION :: placeangle, x1, y1, z1, rannum
 INTEGER :: Receptor_bound_lyn


 ! receptor lyn is bound to
 Receptor_bound_lyn = LynMolecule(iLyn)%Receptor_ID

 ! make sure the current position is set to that of the [boss of] the binding receptor
 LynMolecule(iLyn)%Position =
RecMolecule(RecMolecule(Receptor_bound_lyn)%Boss)%Position

 placeangle = 2*Pi*grnd()

 x1 = cos(placeangle)*Lyn_UnbindRad_RestDimer
 y1 = sin(placeangle)*Lyn_UnbindRad_RestDimer
 z1 = 0


 !this will update lyn position
!!$  CALL
PeriodicBC(LynMolecule(iLyn)%Position(1)+x1,LynMolecule(iLyn)%Position(2)+y1,
&
!!$       LynMolecule(iLyn)%Position(3)+z1, LynMolecule(iLyn)%Position(1), &
!!$       LynMolecule(iLyn)%Position(2),LynMolecule(iLyn)%Position(3))
 CALL
PeriodicBC2(LynMolecule(iLyn)%Position(1)+x1,LynMolecule(iLyn)%Position(2)+y1,
&
     LynMolecule(iLyn)%Position(1), LynMolecule(iLyn)%Position(2))

 ! Update Receptor first
 If (LynMolecule(iLyn)%Itam_site==1) then ! if lyn was on Igalpha
   RecMolecule(Receptor_bound_lyn)%Iga_Lyn = 0 ! update Igalpha on receptor
 else if (LynMolecule(iLyn)%Itam_site==2) then  ! if lyn was on Igbeta
   RecMolecule(Receptor_bound_lyn)%Igb_Lyn = 0 ! update Igbeta on receptor
 End if

 ! update lyn now
 LynMolecule(iLyn)%Receptor_ID = 0
 System_Info%Free_Lyn_Count = System_Info%Free_Lyn_Count + 1
 LynMolecule(iLyn)%Itam_site = 0  ! 0 = unbound, 1 = Igalpha, 2 = Igebta
 LynMolecule(iLyn)%Lyn_site = 0   ! 0 = unbound, 1 = Unique domain, 2 = SH2


!!$  if (System_Info%OutputLevel >=1) &
!!$      write(*,*) 'LynUnBi:  LynID=',iLyn,
'RecID=',Receptor_bound_lyn,'Boss=',RecMolecule(Receptor_bound_lyn)%Boss

  if (System_Info%OutputLevel >=1) &
       write(*,FMT="('T=',f14.8,' LynUnBi LynID=',i3,' RecID=',i3,' Agg=',i3,' Sz=',i0 '
Free Lyn ',i0)") &
       System_Info%Current_Simulation_Time,iLyn,
Receptor_bound_lyn,RecMolecule(Receptor_bound_lyn)%Boss,&
       RecMolecule(Receptor_bound_lyn)%Agg_Size,System_Info%Free_Lyn_Count

END SUBROUTINE LynUnbindReaction

!***********************************************************************
*************************************************************************
*****************


SUBROUTINE ParticleDiffuse(i, DiffSTD)

  ! box boundaries are in ParticleInfo, no need to give them again
  ! the particle number (NP) is also in ParticleInfo, under System_Info%Num_Particles

  USE ParticleInfo
  USE ModelConstants
  USE mtmod

  ! TODO: (optional) update the position of Lyn and Syk bound to receptors being
updated


  IMPLICIT NONE

  ! Declare variables
  INTEGER, INTENT (IN) :: i ! chosen particle
  DOUBLE PRECISION, INTENT (IN) :: DiffSTD ! diffusion standard deviation
  !double precision, intent (in) :: xlimmax,ylimmax,xlimmin,ylimmin, st
  !double precision, intent (in) :: st
  INTEGER :: ParticleSpecies, BoundBud, domainnum,
Initial_domain_check,Suspected_domain,k
  DOUBLE PRECISION :: r1, r2, w1, w2, x1, y1, rannum, escape_probability
  double precision :: NewCoord(3) ! tentative new position of the particle (replaces
MoveDistance)
  integer :: DomainEscape ! will be true (+1) if a comain escape occurs
  integer :: NewDomain, EscapeAttempt,DomainChange
```

```
!!! randomly make a trajectory for particles using mtmod.f90 for random numbers !!!
  !! Generate random number & Normally distribute random number !
http://www.taygeta.com/random/gaussian.html
  !* Generate x move
  r1=2*grnd()-1
  r2=2*grnd()-1
  ! Check unit circle, if not in reject and try again
  w1=r1*r1+r2*r2
  DO WHILE (w1 > 1)
    ! Generate random number again
    r1=2*grnd()-1
    r2=2*grnd()-1
    ! Unit circle check
    w1=r1*r1+r2*r2
  END DO
  w2=sqrt((-2*log(w1))/w1)
  ! Normally distributed random # for distance
  x1=r1*w2

  ! Generate y move
  r1=2*grnd()-1
  r2=2*grnd()-1
  ! Check unit circle, if not in reject and try again
  w1=r1*r1+r2*r2
  DO WHILE (w1 > 1)
    ! Generate random number again
    r1=2*grnd()-1
    r2=2*grnd()-1
    ! Unit circle check
    w1=r1*r1+r2*r2
  END DO
  w2=sqrt((-2*log(w1))/w1)
  y1=r2*w2

  ! Account for diffusion coefficient based on species type
  y1=y1*DiffSTD
  x1=x1*DiffSTD



  ! New position (tentative for now)
  ! Before dealing with the domains, we need to satisfy the periodic BC
  call PeriodicBC2(&
      RecMolecule(i)%Position(1)+x1,RecMolecule(i)%Position(2)+y1,&
      NewCoord(1),NewCoord(2));
```

```
 NewCoord(3)=0 ! no move off the membrane for a receptor

 ! Check if receptor is in free space and whether it is anticipated to move ---> into a
domain OR another free space

 Initial_domain_check = RecMolecule(i)%Domain   ! Could be domains --> 0, 1, 2 ,3, 4,
5

 DomainEscape  = 0; ! assume no domain change
 EscapeAttempt = 0; !
 DomainChange  = 0; ! zero if domain ID does not change in the end

 ! 4 possibilities:
 ! 1.     domain -> free space   : Reflective BC, if escaped then periodic BC
 ! 2.     domain -> domain       : No BC
 ! 3.  free space -> domain      : No boundary check required
 ! 4.  free space -> free space  : Periodic BC


 if (RecMolecule(i)%Domain>0) then
   ! Case 1+2 .. particle is in a domain before the move

   if (InDomain(NewCoord(1:2),Initial_domain_check)) then
     ! Case 2: stays in the domain, all is well
   else
     ! wants to leave the domain
     ! used to call DomainEscape() here
     EscapeAttempt = 1

     if(grnd()<=EscapeProb) then
       ! Case 1: move will be accepted - escape success
       DomainEscape = 1
     else
       ! move is rejected, particle should be reflected off the boundary
       NewCoord(1:2) = NewCoord(1:2) - 2*[x1,y1];
       ! paranoid check on whether the reflected move would constitute an escape
       if(InDomain(NewCoord(1:2),Initial_domain_check)) then
         ! the reflected move is in the domain, hence accepted
       else
         ! the particle is stuck in a narrow section
         NewCoord = RecMolecule(i)%Position
       endif
     endif

     ! at this point, we have a definite new position (NewCoord)
     ! which is verified within the simulation box (did BC first thing)
```

```
      ! the particle may have escaped its initial domain in which case it is
      ! assumed in free space ; we need to take care of the case when this
      ! new position immediately puts it into another domain

    endif! new position in domain or not

    ! at this point:
    ! NewCoord has the accepted new position
    ! DomainEscape = 1 if and only if the particle was allowed to leave its initial domain
    ! EscapeAttempt = 1 iff the proposed move was to leave (accepted or not)

    if(DomainEscape==1) then
      DomainChange=1
      ! update domain info to free
      where(RecMolecule(:)%Boss==i)
        RecMolecule(:)%Domain = 0
      end where
    endif


    ! all of the above was for when the particle was initially in a domain
  endif

  ! NewCoord is valid now whether the particle was free or not and escaped or not
  ! time to update the position for this receptor and its subordinates
  where(RecMolecule(:)%Boss==i)
    RecMolecule(:)%Position(1)=NewCoord(1)
    RecMolecule(:)%Position(2)=NewCoord(2);
  end where
  RecMolecule(:)%Position(3)=0 ! receptors never leave the membrane

  ! still need to check if the new position puts the particle in a NEW domain
  if(Initial_domain_check == 0 .or. DomainEscape > 0 ) then

    NewDomain=0;

    do k=1,NumDomains

      if ((NewCoord(1) >= Dom(k)%Xlim(1))  .AND. (NewCoord(1)<=
Dom(k)%Xlim(2))  &
          .AND.(NewCoord(2) >= Dom(k)%Ylim(1)) .AND.
(NewCoord(2)<=Dom(k)%Ylim(2)))  then

        if(InDomain(NewCoord(1:2),k)) NewDomain=k

      endif
    end do

    if (NewDomain>0) then

      if (Initial_domain_check > 0 .and. NewDomain == Initial_domain_check) then
        write(*,*) 'Error - domain escape and not'
      else
        ! legit trapping in NewDomain
        DomainChange = 1
        ! update domain info to free
        where(RecMolecule(:)%Boss==i)
          RecMolecule(:)%Domain = NewDomain
        end where
      endif

    endif

  endif ! if initially free or escaped

  ! now
  ! DomainEscape=1 if escape from a domain
  ! DomainChange=1 if escaped and/or entered a new domain
  ! NewDomain is correct in both cases above


  if (DomainChange==1) then

    DomainParticleCount(Initial_domain_check) =
DomainParticleCount(Initial_domain_check) - RecMolecule(i)%Agg_Size
    DomainParticleCount(NewDomain) =  DomainParticleCount(NewDomain) +
RecMolecule(i)%Agg_Size

    if (System_Info%OutputLevel >=1) &
       write(*,FMT="(&
       'T=',f14.8,' DomChg  Rec1 =',i3,&
       ' Dom ',I1,' --> ',I1,&
       ' Boss ',i3,' AggSize ',i2,' DomCounts:', 5(i3,' '))") &
       System_Info%Current_Simulation_Time,i,Initial_domain_check,NewDomain,&
       RecMolecule(i)%Boss,RecMolecule(i)%Agg_Size,&
       DomainParticleCount(1:NumDomains)
  endif


  CALL BindReaction(i)

END SUBROUTINE ParticleDiffuse
```

```
!************************************************************************
*************************************************************************
************
SUBROUTINE ReflectiveBC (i,x1,y1)

  USE ParticleInfo
  USE mtmod
  USE ModelConstants

  IMPLICIT NONE


  DOUBLE PRECISION, INTENT(IN) :: x1, y1
  INTEGER, INTENT(IN) :: i ! Current receptor
  DOUBLE PRECISION :: x, y ! Proposed move


  x=RecMolecule(i)%Position(1)-x1
  y=RecMolecule(i)%Position(2)-y1

  RecMolecule(i)%Position(1) = x
  RecMolecule(i)%Position(2) = y

END SUBROUTINE  ReflectiveBC


!************************************************************************
*************************************************************************
************

SUBROUTINE PeriodicBC2(x,y,xnew,ynew)

  USE ParticleInfo

  ! 2-d verions ot be used for membrane bound species (Rec, Lyn)

  IMPLICIT NONE

  DOUBLE PRECISION, INTENT(IN) :: x, y ! Proposed new x, Proposed new y
  DOUBLE PRECISION, INTENT(OUT) :: xnew, ynew ! New BC satisfied coordinates

  DOUBLE PRECISION ::  xmax, ymax, ymin, xmin

  ! Limits on x and y coordinates defined by system boundaries
  xmin=System_Info%SimSpace_Boundary(3)
  xmax=System_Info%SimSpace_Boundary(1)
  ymin=System_Info%SimSpace_Boundary(4)
  ymax=System_Info%SimSpace_Boundary(2)


  ! Check & Apply Periodic Boundary Condition

  ! Check x move
  IF (x < xmin) THEN ! Add width of box
    xnew=x+(xmax-xmin)
  ELSE IF (x > xmax) THEN ! Subtract width of box
    xnew=x-(xmax-xmin)
  ELSE ! remains unchanged
    xnew=x
  END IF

  ! Check y move
  IF (y < ymin) THEN ! Add length of box
    ! Define Y New Coordinate
    ynew=y+(ymax-ymin)
  ELSE IF (y > ymax) THEN ! Subtract length of box
    ! Define Y New Coordinate
    ynew=y-(ymax-ymin)
  ELSE ! remains unchanged
    ! Define Y New Coordinate
    ynew=y
  END IF

END SUBROUTINE PeriodicBC2

SUBROUTINE PeriodicBC3(x,y,z,xnew,ynew,znew)

  ! to be used for 3d species (Syk)
  ! NOTE: (1) BC in the z-direction are REFLECTIVE
  !       (2) the simulation boundaries in z are 0 (top) and -Depth (bottom)

  USE ParticleInfo

  IMPLICIT NONE

  DOUBLE PRECISION, INTENT(IN) :: x, y, z ! Proposed new x, Proposed new y
  DOUBLE PRECISION, INTENT(OUT) :: xnew, ynew, znew ! New BC satisfied
coordinates
```

```
DOUBLE PRECISION ::  xmax, ymax, ymin, xmin, zmin, zmax

! Limits on x and y coordinates defined by system boundaries

xmin=System_Info%SimSpace_Boundary(3)
xmax=System_Info%SimSpace_Boundary(1)
ymin=System_Info%SimSpace_Boundary(4)
ymax=System_Info%SimSpace_Boundary(2)

zmin = System_Info%SimSpace_Boundary(5)
zmax = System_Info%SimSpace_Boundary(6)

! Check & Apply Periodic Boundary Condition

! Check x move
IF (x < xmin) THEN ! Add width of box
  xnew=x+(xmax-xmin)
ELSE IF (x > xmax) THEN ! Subtract width of box
  xnew=x-(xmax-xmin)
ELSE ! remains unchanged
  xnew=x
END IF

! Check y move
IF (y < ymin) THEN ! Add length of box
  ! Define Y New Coordinate
  ynew=y+(ymax-ymin)
ELSE IF (y > ymax) THEN ! Subtract length of box
  ! Define Y New Coordinate
  ynew=y-(ymax-ymin)
ELSE ! remains unchanged
  ! Define Y New Coordinate
  ynew=y
END IF


! Check z move only if z > 0
! the z (vertical) direction must have REFLECTIVE BC

if (z/=0) then

  ! reflect by zmin
  if (z<zmin) then
    znew=zmin + (zmin - z)
  end if
```

```
    if (z>zmax) then
      znew=zmax - (z-zmax)
    end if

  endif

END SUBROUTINE PeriodicBC3

!*****************************************************************
******************************************************************
************


SUBROUTINE BindReaction(i)

  ! TODO : add collision check and binding with Lyn and Syk

  ! the particle number (NP) and system time (st) are available in ParticleInfo,
  ! under System_Info%Current_Simulation_Time and System_Info%Num_Particles

  USE ParticleInfo
  USE ModelConstants
  USE mtmod

  IMPLICIT NONE

  INTEGER, INTENT(IN) ::  i ! specific particle

  ! LOGICAL :: Reaction -- using System_Info%Reaction
  INTEGER :: Bond_Count, Bond_Count_i, k, Bond_count_k
  INTEGER :: Agg1(System_Info%Num_Particles),
Agg2(System_Info%Num_Particles),&
      AggT(System_Info%Num_Particles), size, next, d, f
  INTEGER :: NewBoss, NewAggSize, CurrentParticle, ip, c
  INTEGER :: OldAggSize_i, OldAggSize_k
  DOUBLE PRECISION :: distsq
  double precision :: BeginTime_i, BeginTime_k ! begin times of the merging aggregates



  if (System_Info%OutputLevel >= 3) write(*,*) &
      'BindReaction begin: st:',System_Info%Current_Simulation_Time,&
      ' NP:', System_Info%Num_Particles,' i=',i, &
      ' size:', RecMolecule(i)%Agg_Size
```

```
 System_Info%Reaction = .FALSE.
 !Reaction = .FALSE.


 ! Check for reaction
 DO k = 1,System_Info%Num_Particles ! loop over particles (receptor monomers)

   ! Check if one of available monomers is close enough to react
   IF (k /= i .AND. RecMolecule(k)%Boss == k ) THEN

     ! Calculate distance between particle of interest and compared particle
     distsq=sqrt((RecMolecule(k)%Position(1)-RecMolecule(i)%Position(1))**2+&
        (RecMolecule(k)%Position(2)-RecMolecule(i)%Position(2))**2)

     Bond_count_i = RecMolecule(i)%Bond
     Bond_count_k = RecMolecule(k)%Bond

     IF (distsq <= BindRad_Dimer) THEN  !LR-RL

       System_Info%Reaction = .TRUE.

       OldAggSize_k = RecMolecule(k)%Agg_Size
       OldAggSize_i = RecMolecule(i)%Agg_Size
       NewAggSize  = RecMolecule(i)%Agg_Size + RecMolecule(k)%Agg_Size

       BeginTime_k = RecMolecule(k)%LastOnOffTime
       BeginTime_i = RecMolecule(i)%LastOnOffTime



       System_Info%Num_Aggregates = System_Info%Num_Aggregates - 1

       System_Info%AggSizeCount( OldAggSize_k) =
System_Info%AggSizeCount(OldAggSize_k) - 1
       System_Info%AggSizeCount( OldAggSize_i) =
System_Info%AggSizeCount(OldAggSize_i) - 1
       System_Info%AggSizeCount(NewAggSize) =
System_Info%AggSizeCount(NewAggSize) + 1

       if (System_Info%OutputLevel >=3) &
          write(*,*) 'BindReaction - success ',i,k


     END IF

     f = RecMolecule(i)%Agg_Size ! same as OldAggSize_i
```

```
if(System_Info%Reaction)then

  ! Go down chain k
  CurrentParticle=k ! k is the binding partner we just found
  AggT(1)=CurrentParticle
  d = RecMolecule(k)%Agg_Size ! same as OldAggSize_k
  DO ip=2,OldAggSize_k
    CurrentParticle=RecMolecule(CurrentParticle)%BoundRec_2
    AggT(ip) = CurrentParticle
  END DO

  ! Go down chain i
  CurrentParticle=i ! i is the originally chosen particle
  Agg1(1)=CurrentParticle
  DO ip=2,OldAggSize_i
    CurrentParticle=RecMolecule(CurrentParticle)%BoundRec_2
    Agg1(ip) = CurrentParticle
  END DO

  next=CurrentParticle

  ! Make the bond:
  ! head of partner chain (k)
  ! connected to tail of incoming chain (next)
  RecMolecule(k)%BoundRec_1 = next ! head of partner chain front link
  RecMolecule(next)%BoundRec_2 = k ! tail of incoming chain back link


  ! increment bond counts
  RecMolecule(k)%Bond = RecMolecule(k)%Bond + 1
  RecMolecule(next)%Bond = RecMolecule(next)%Bond + 1

  ! Particle(k)%Position(:)=Particle(i)%Position(:) ! position of partner particle -- why only this one ???

  !Particle(i)%Agg_Size = Particle(i)%Agg_Size+Particle(k)%Agg_Size ! agg size update -- why only this one ???

  ! update the entire merged aggregate
  WHERE (RecMolecule(:)%Boss == i .OR. RecMolecule(:)%Boss == k)
    RecMolecule(:)%Position(1) = RecMolecule(i)%Position(1)
    RecMolecule(:)%Position(2) = RecMolecule(i)%Position(2)
    RecMolecule(:)%Position(3) = RecMolecule(i)%Position(3)
```

```
         RecMolecule(:)%Agg_Size = NewAggSize
         RecMolecule(:)%Boss = i
         RecMolecule(:)%LastOnOffTime = System_Info%Current_Simulation_Time
       END WHERE

!!$          if (System_Info%OutputLevel>=1) then
!!$             write(*,*) 'Time',System_Info%Current_Simulation_Time,&
!!$                  'RecBind ','RecID ',i,k,&
!!$                  'OldAggSizes ', OldAggSize_i, OldAggSize_k, &
!!$                  'NewAggSize ',NewAggSize

       if (System_Info%OutputLevel >=1) &
          write(*,FMT="(&
          'T=',f14.8,' RecBind Rec1 =',i3,' Rec2 =',i3,&
          ' Aggs [',I3,',',I3,'] --> [',I3,']',&
          ' Size ',i2,'+',i2,'=',i2,' AggCount ',i3,' Dom ',i1,' ',i1)") &
          System_Info%Current_Simulation_Time,k,next,i,k,i,&
          OldAggSize_i,OldAggSize_k,NewAggSize,
System_Info%Num_Aggregates,&
          RecMolecule(i)%Domain,RecMolecule(k)%Domain

     endif ! if a reaction occurs


   end if ! if k is a boss

   if (System_Info%Reaction) exit ! break the loop / so only one reaction per mini update

 end do ! loop over possible reaction partners (k)

 if(System_Info%Reaction) then


   ! check the aggregate by walking down the list
   CurrentParticle=i;
   do
     if (System_Info%OutputLevel >= 2) write(*,1003) CurrentParticle, &
        RecMolecule(CurrentParticle)%BoundRec_1,
RecMolecule(CurrentParticle)%BoundRec_2,&
        RecMolecule(CurrentParticle)%Bond,&
        RecMolecule(CurrentParticle)%Agg_Size,
RecMolecule(CurrentParticle)%Boss,&
        RecMolecule(CurrentParticle)%Position(1:2)
1003    format('    ',6I5,2F8.3)
     if( RecMolecule(CurrentParticle)%BoundRec_2==0) exit
     CurrentParticle = RecMolecule(CurrentParticle)%BoundRec_2
```

```
    end do

  else

    if (System_Info%OutputLevel >=3) write(*,*) 'BindReaction -- no reaction this time'

  endif

  if (System_Info%OutputLevel >= 3) WRITE(*,*) 'BindReaction end'

  ! output for "dimer lifetime" record
  if(System_Info%Reaction)then

    ! we need to output enough information to completely define the aggregates that
    ! are ending with this reaction

    ! Metadata (for each ending aggregate): Length, Begin and end times, Boss ID
    ! List of member receptors beginning with the Boss, on a single line

    WRITE(4,*) 'Bind', System_Info%Current_Simulation_Time,&
        'Ksize',OldAggSize_k,'kagg',    AggT(1:OldAggSize_k),&
        'Isize',OldAggSize_i,'oldiagg', Agg1(1:OldAggSize_i)

    ! post mortem aggregate entry: Length, Boss, DescendantBoss1, DescendantBoss2,
BeginTime, EndTime, <Receptor list>
    write(4,*), OldAggSize_i,i,k,0, BeginTime_i,
System_Info%Current_Simulation_Time, Agg1(1:OldAggSize_i)
    write(4,*), OldAggSize_k,i,k,0, BeginTime_k,
System_Info%Current_Simulation_Time, AggT(1:OldAggSize_k)


    !WRITE(4,*) 'Bind', System_Info%Current_Simulation_Time, i, k, OldAggSize1,
OldAggSize2, NewAggSize
  end if


END SUBROUTINE BindReaction


!*********************************************************************
**********************************************************************
************

SUBROUTINE UnbindReaction(i)

  ! i identifies the aggregate the breaks (the boss of it is receptor i)
  USE ParticleInfo
  USE ModelConstants
  USE mtmod

  IMPLICIT NONE

  integer, intent(in) :: i

  DOUBLE PRECISION :: placeangle, x1, y1, z1, rannum
  INTEGER :: LigandCount, domainnum, BoundBuddy
  INTEGER :: BondToBreak, BondToBreakPlusOne, Break_Particle_1,
Break_Particle_2, CurrentParticle, ip
  INTEGER :: Agg2(System_Info%Num_Particles),
Agg3(System_Info%Num_Particles),ic
  integer :: OldAggSize,NewAggSize1,NewAggSize2, NewBoss
  integer :: OldDomain,NewDomain,iDom
  double precision :: BeginTime
  double precision :: OldPosition(3),NewPosition(3)

  ! INTEGER, PARAMETER :: OutputLevel = 0


  if (System_Info%OutputLevel >= 2) write(*,1002) i,RecMolecule(i)%Agg_Size
1002 format('UnBindReaction Input: Agg:',I3,' size ',I3);

  BeginTime = RecMolecule(i)%LastOnOffTime

  ! check the aggregate by walking down the list
  CurrentParticle=i;
  do
    if (System_Info%OutputLevel >= 2) write(*,1003) CurrentParticle, &
        RecMolecule(CurrentParticle)%BoundRec_1,
RecMolecule(CurrentParticle)%BoundRec_2,&
        RecMolecule(CurrentParticle)%Bond,&
        RecMolecule(CurrentParticle)%Agg_Size, RecMolecule(CurrentParticle)%Boss,&
        RecMolecule(CurrentParticle)%Position
1003 format('    ',6I5,2F8.3)
    if(RecMolecule(CurrentParticle)%BoundRec_2==0) exit
    CurrentParticle = RecMolecule(CurrentParticle)%BoundRec_2
  end do

  !Break_bond = floor(rannum*(Agg_Size-1))
  !k = Break_bond

  ! This will give me bond to break
  rannum = grnd()
  BondToBreak = ceiling(rannum*(RecMolecule(i)%Agg_Size-1))

  if (System_Info%OutputLevel >= 2) write(*,*) 'UnBindReaction: Agg ',i,' size
',RecMolecule(i)%Agg_Size, ' break at ', BondToBreak
  !write(*,*) 'Breaking bond',BondToBreak,' of ', RecMolecule(i)%Agg_Size-1

  ! walk down the chain k steps

  CurrentParticle=i
  Agg2(1)=CurrentParticle
  if (System_Info%OutputLevel >= 2) write(*,*) '   Chain start ',CurrentParticle,&
      ' bonds: ',RecMolecule(CurrentParticle)%BoundRec_1,
RecMolecule(CurrentParticle)%BoundRec_2
  DO ip=2,RecMolecule(i)%Agg_Size
    CurrentParticle=RecMolecule(CurrentParticle)%BoundRec_2
    Agg2(ip) = CurrentParticle
    if (System_Info%OutputLevel >= 2) write(*,*) '          next ',CurrentParticle,&
        ' bonds: ',RecMolecule(CurrentParticle)%BoundRec_1,
RecMolecule(CurrentParticle)%BoundRec_2
    !WRITE(*,*) 'LOOP'
  END DO


  ! identifiers of the particles where the chain breaks
  Break_Particle_1 = Agg2(BondToBreak)
  BondToBreakPlusOne = BondToBreak + 1
  Break_Particle_2 = Agg2(BondToBreakPlusOne)

  OldAggSize = RecMolecule(i)%Agg_Size
  NewAggSize1 = BondToBreak
  NewAggSize2 = OldAggSize - BondToBreak
  NewBoss = Agg2(BondToBreakPlusOne)

  !first chain is Agg2(1:NewAggSize1)
  !second chain is Agg2(NewAggSize1+1 : OldAggSize) or
Agg2(BondToBreakPlusOne:OldAggSize)

  ! Deal with chain 1
  ! Update BoundRec_2
  RecMolecule(Break_Particle_1)%BoundRec_2 = 0
  ! Update Bond (0 or 1)
  IF (RecMolecule(Break_Particle_1)%BoundRec_1 == 0) THEN
    RecMolecule(Break_Particle_1)%Bond= 0
  ELSE
    RecMolecule(Break_Particle_1)%Bond= 1
  END IF

  ! first piece
  !Particle(Agg2(1:NewAggSize1))%Boss=i ! unnecessary
  !Update Aggsize of first aggregate
  RecMolecule(Agg2(1:NewAggSize1))%Agg_Size=NewAggSize1



  !Deal with chain 2
  RecMolecule(Break_Particle_2)%BoundRec_1 = 0  !update boundrec_1
  IF (RecMolecule(Break_Particle_2)%BoundRec_2 == 0) THEN  ! update bond
    RecMolecule(Break_Particle_2)%Bond = 0
  ELSE
    RecMolecule(Break_Particle_2)%Bond = 1
  END IF



  ! second piece
  RecMolecule(Agg2(BondToBreakPlusOne:OldAggSize))%Boss = NewBoss  ! update
Boss
  RecMolecule(Agg2(BondToBreakPlusOne:OldAggSize))%Agg_Size = NewAggSize2
!update agg size

  ! figure out the ejected piece's position
  OldPosition = RecMolecule(Break_Particle_2)%Position
  OldDomain   = RecMolecule(Break_Particle_2)%Domain
  NewDomain = OldDomain ! not really necessary, it may change only if OldDomain=0

  ! Add new x and y coordinates
  placeangle=2*Pi*grnd() ! Pick a number between 0 and 2*Pi

  x1=cos(placeangle)*UnbindRad_RestDimer
  y1=sin(placeangle)*UnbindRad_RestDimer
  z1= 0

  ! impose periodic BC on the ejected particle's proposed position
  ! call
PeriodicBC(OldPosition(1)+x1,OldPosition(2)+y1,OldPosition(3),NewPosition(1),NewP
osition(2),NewPosition(3))
  call
PeriodicBC2(OldPosition(1)+x1,OldPosition(2)+y1,NewPosition(1),NewPosition(2))

  ! now check if there is a domain escape involved
  if(OldDomain > 0) then
```

```
    ! if this move goes outside the domain, try the opposite direction
    if(.not. InDomain(NewPosition(1:2),OldDomain)) &
        call PeriodicBC2(OldPosition(1)-x1,OldPosition(2)-
y1,NewPosition(1),NewPosition(2))
    ! if this move goes out, try the perpendicular direction
    if(.not.InDomain(NewPosition(1:2),OldDomain)) &
        call PeriodicBC2(OldPosition(1)-
y1,OldPosition(2)+x1,NewPosition(1),NewPosition(2))
    ! if this move goes out, try the opposite perpendicular direction
    if(.not.InDomain(NewPosition(1:2),OldDomain)) &
        call PeriodicBC2(OldPosition(1)+y1,OldPosition(2)-
x1,NewPosition(1),NewPosition(2))
    ! if this move still goes out, give up
    if(.not.InDomain(NewPosition(1:2),OldDomain)) &
        NewPosition = OldPosition
  else
    ! if particle was initially free, it may end up in a domain
    NewDomain = 0
    do iDom=1,NumDomains
      if(InDomain(NewPosition(1:2),iDom)) NewDomain=iDom
    end do
  endif

  ! record the ejected position
  RecMolecule(Break_Particle_2)%Position = NewPosition

  ! if trapping occurred
  if(OldDomain==0 .and. NewDomain>0) then
    RecMolecule(Break_Particle_2)%Domain = NewDomain
    DomainParticleCount(NewDomain) = DomainParticleCount(NewDomain) +
NewAggSize2
  endif

  ! update Position in 2nd chain
  RecMolecule(Agg2(BondToBreakPlusOne:OldAggSize))%Position(1) =
RecMolecule(Break_Particle_2)%Position(1)
  RecMolecule(Agg2(BondToBreakPlusOne:OldAggSize))%Position(2) =
RecMolecule(Break_Particle_2)%Position(2)
  RecMolecule(Agg2(BondToBreakPlusOne:OldAggSize))%Position(3) =
RecMolecule(Break_Particle_2)%Position(3)
  RecMolecule(Agg2(BondToBreakPlusOne:OldAggSize))%Domain      =
RecMolecule(Break_Particle_2)%Domain

  ! update the last reaction time for all receptors in the original aggregate
  RecMolecule(Agg2(1:OldAggSize))%LastOnOffTime =
System_Info%Current_Simulation_Time
```

```
  System_Info%Num_Aggregates = System_Info%Num_Aggregates + 1

  System_Info%AggSizeCount(OldAggSize)  = 
System_Info%AggSizeCount(OldAggSize) - 1
  System_Info%AggSizeCount(NewAggSize1) = 
System_Info%AggSizeCount(NewAggSize1) + 1
  System_Info%AggSizeCount(NewAggSize2) = 
System_Info%AggSizeCount(NewAggSize2) + 1

  WRITE(4,*) 'UnBi', System_Info%Current_Simulation_Time,&
      i,NewBoss,OldAggSize,NewAggSize1,NewAggSize2

  ! post mortem aggregate entry: Length, Boss, DescendantBoss1, DescendantBoss2, 
BeginTime, EndTime, <Receptor list>
  write(4,*) OldAggSize, i, i, NewBoss,&
      BeginTime, System_Info%Current_Simulation_Time, Agg2(1:OldAggSize)

  if (System_Info%OutputLevel >= 1) &
      write(*,1004)  System_Info%Current_Simulation_Time,&
      Break_Particle_1, Break_Particle_2,&
      i,i,NewBoss,OldAggSize,NewAggSize1,NewAggSize2, 
System_Info%Num_Aggregates, &
      OldDomain, NewDomain

1004 format('T=',F14.8,' RecUnBi Rec1 =',i3,' Rec2 =',i3,&
      ' Aggs [',I3,'] --> [',I3,',',I3,']',&
      ' Size ',I2,'=',I2,'+',I2,' AggCount ',I3,' Dom ',i1,' ',i1)

  if (System_Info%OutputLevel >= 2) write(*,*) '    One   '
  ! check the aggregate by walking down the list
  CurrentParticle=i;
  do
    if (System_Info%OutputLevel >= 2) write(*,1003) CurrentParticle, &
        RecMolecule(CurrentParticle)%BoundRec_1, 
RecMolecule(CurrentParticle)%BoundRec_2,&
        RecMolecule(CurrentParticle)%Bond,&
        RecMolecule(CurrentParticle)%Agg_Size, RecMolecule(CurrentParticle)%Boss,&
        RecMolecule(CurrentParticle)%Position
    if( RecMolecule(CurrentParticle)%BoundRec_2==0) exit
    CurrentParticle = RecMolecule(CurrentParticle)%BoundRec_2
  end do


  if (System_Info%OutputLevel >= 2) write(*,*) '    Two   '
  ! check the aggregate by walking down the list
  CurrentParticle=Break_Particle_2;
  do
    if (System_Info%OutputLevel >= 2) write(*,1003) CurrentParticle, &
        RecMolecule(CurrentParticle)%BoundRec_1,
RecMolecule(CurrentParticle)%BoundRec_2,&
        RecMolecule(CurrentParticle)%Bond,&
        RecMolecule(CurrentParticle)%Agg_Size, RecMolecule(CurrentParticle)%Boss,&
        RecMolecule(CurrentParticle)%Position
    if( RecMolecule(CurrentParticle)%BoundRec_2==0) exit
    CurrentParticle = RecMolecule(CurrentParticle)%BoundRec_2
  end do

  if (System_Info%OutputLevel >= 2) write(*,*) '  done with this unbinding '


  ! Record time of undimerization
  !RecMolecule(i)%DimOffTime=st
  !Particle(BoundBuddy)%DimOffTime=st
  ! Record who was in the dimer
  !Particle(i)%PrevBuddy=BoundBuddy
  !Particle(BoundBuddy)%PrevBuddy=i
  ! Record dimer lifetime and type
  !WRITE(4,*) Particle(i)%Lifetime, LigandCount, Particle(i)%DimerOnTime, &
  !                   &Particle(i)%DimOffTime, i, boundbuddy ! Dimer_liftime
Ligands/Dimer, dimer on time, dimer off time, receptor, bound receptor
  ! Reset dimer lifetime
  !Particle(i)%Lifetime=0
  !Particle(BoundBuddy)%Lifetime=0
  ! Reset time of undimerization
  !Particle(i)%DimOffTime=0
  !Particle(BoundBuddy)%DimOffTime=0
  ! Reset time of dimerization
  !Particle(i)%DimerOnTime=0
  !Particle(BoundBuddy)%DimerOnTime=0
  ! Reset Time to Phosphorylation
  !Particle(i)%PhosTime=0
  !Particle(BoundBuddy)%PhosTime=0
  ! Reset Phosphorylation Event
  !Particle(i)%PhosEvent=0
  !Particle(BoundBuddy)%PhosEvent=0
  ! Seperate dimer in to orignal monomer species
  !Particle(i)%Species=Particle(i)%OriginalSpecies
  !Particle(BoundBuddy)%Species=Particle(BoundBuddy)%OriginalSpecies


  ! Update species back to monomers and erase partners
  !Particle(i)%DomPartner = 0
  !Particle(BoundBuddy)%DomPartner = 0
  !       Particle(BoundBuddy)%BoundBuddy = 0
  !       Particle(i)%BoundBuddy = 0
  !       BoundBuddy=0

  if (System_Info%OutputLevel >= 2)  write(*,*) 'UnBindReaction end'


END SUBROUTINE UnbindReaction

!****************************************************************
*****************************************************************
************

SUBROUTINE SykDiffuse (iSyk, SykDiffuse_std)


  USE ParticleInfo
  USE ModelConstants
  USE mtmod

  IMPLICIT NONE

  ! Declare variables
  INTEGER, INTENT (IN) :: iSyk ! chosen particle
  DOUBLE PRECISION, INTENT (IN) :: SykDiffuse_std ! diffusion standard deviation
  DOUBLE PRECISION :: r1, r2, r3, w1, w2, w3, x1, y1, z1, rannum

  if (System_Info%Reaction .or. System_Info%OutputLevel >=2) write(*,*) 'SykDiffuse
begin -- SykID ',iSyk, 'Coords: ',&
       SykMolecule(iSyk)%Position

  Syk_DiffCall_Count = Syk_DiffCall_Count + 1

!!! randomly make a trajectory for particles using mtmod.f90 for random numbers !!!
  !! Generate random number & Normally distribute random number !
http://www.taygeta.com/random/gaussian.html
  !* Generate x move
  r1=2*grnd()-1
  r2=2*grnd()-1
  ! Check unit circle, if not in reject and try again
  w1=r1*r1+r2*r2
  DO WHILE (w1 > 1)
    ! Generate random number again
    r1=2*grnd()-1
    r2=2*grnd()-1
    ! Unit circle check
    w1=r1*r1+r2*r2
  END DO
  w2=sqrt((-2*log(w1))/w1)
  ! Normally distributed random # for distance
  x1=r1*w2

  ! Generate y move
  r1=2*grnd()-1
  r2=2*grnd()-1
  ! Check unit circle, if not in reject and try again
  w1=r1*r1+r2*r2
  DO WHILE (w1 > 1)
    ! Generate random number again
    r1=2*grnd()-1
    r2=2*grnd()-1
    ! Unit circle check
    w1=r1*r1+r2*r2
  END DO
  w2=sqrt((-2*log(w1))/w1)
  y1=r2*w2


  !Generate z move
  r1=2*grnd()-1
  r2=2*grnd()-1
  ! Check unit circle, if not in reject and try again
  w1=r1*r1+r2*r2
  DO WHILE (w1 > 1)
    ! Generate random number again
    r1=2*grnd()-1
    r2=2*grnd()-1
    ! Unit circle check
    w1=r1*r1+r2*r2
  END DO
  w2=sqrt((-2*log(w1))/w1)
  z1=r2*w2

  ! Account for diffusion coefficient based on species type
  y1=y1*SykDiffuse_std
  x1=x1*SykDiffuse_std
  z1=z1*SykDiffuse_std
```

```
  if (System_Info%Reaction .or. System_Info%OutputLevel >=2) write(*,*) 'SykDiffuse
- call PBC'

  ! call periodic boundary condition
  CALL PeriodicBC3( &
      SykMolecule(iSyk)%Position(1)+x1,SykMolecule(iSyk)%Position(2)+y1,
SykMolecule(iSyk)%Position(3)+z1, &
      SykMolecule(iSyk)%Position(1),
SykMolecule(iSyk)%Position(2),SykMolecule(iSyk)%Position(3))

  if (System_Info%Reaction .or. System_Info%OutputLevel >=2) write(*,*) 'SykDiffuse
SykID ',iSyk, 'New Coords: ',&
      SykMolecule(iSyk)%Position(1:3)

  ! Check for (dimerization) reactions
  if(abs(SykMolecule(iSyk)%Position(3))<SykLayerDepth) &
      CALL SykBind(iSyk)

  if (System_Info%Reaction .or. System_Info%OutputLevel >=2) write(*,*) 'SykDiffuse
end ',iSyk

  if (System_Info%Reaction) then
    Syk_Diff_Reaction = Syk_Diff_Reaction + 1
  else
    Syk_Diff_NoReaction = Syk_Diff_NoReaction + 1
  end if

END SUBROUTINE SykDiffuse

!**************************************************************
******************************************************************
************

SUBROUTINE SykBind (iSyk)


  USE ParticleInfo
  USE ModelConstants
  USE mtmod

  IMPLICIT NONE

  INTEGER, INTENT(IN)::iSyk
  INTEGER, DIMENSION(MaxAgg) :: Syk_Agg=0 ! list of receptors in a given
aggregate
```

```
  real, dimension(MaxAgg) :: IgABindProb, IgBBindProb
  real, dimension(2*MaxAgg) :: SykBindProbVec
  real :: CumSum=0

  DOUBLE PRECISION :: distsq, rannum
  INTEGER :: j, u, m_syk, k, i, rv_syk,cv_syk,ip,h, i_syk

  integer, dimension(System_Info%Num_Particles) :: BossList
  integer :: BossCount2

  integer :: AggRecCount ! number of receptors on the current aggregate

  ! counters for partners (eligible aggregates), docking sites (one- and two-SH2)
  integer :: BossCount, EligibleAggCount=0 ! count per each sub call
  integer :: ThisAggEligible=0 ! by agg
  integer, dimension(2) :: AggPhosCount=0, SysPhosCount=0 ! count per agg then add up
  double precision :: MinDistNow ! closest square dist to an eligible aggregate

  if (System_Info%OutputLevel >=2) write(*,*) 'SykBind begin -- SykID ',iSyk

  ! counters..
  Syk_BindCall_Count = Syk_BindCall_Count + 1 ! counts each call to this sub
  BossCount = 0 ! count bosses in this sub call
  EligibleAggCount=0
  SysPhosCount=0
  MinDistNow=100

  ! TODO: try to identify the receptors that have docking sites, and their bosses
  !      those are the only receptors that should be checked in the main loop below
  !

!!$  BossCount2 = count(RecMolecule(:)%Iga_Phos > 0 .and.
RecMolecule(:)%Iga_Lyn==0 .and. RecMolecule(:)%Iga_Syk==0)
!!$  BossList = pack(RecMolecule(:)%Boss, RecMolecule(:)%Iga_Phos > 0 .and.
RecMolecule(:)%Iga_Lyn==0 .and. RecMolecule(:)%Iga_Syk==0)
!!$
!!$  if(BossCount2==10) then
!!$     do j=1,10
!!$        write(*,*) 'BOSSLIST!! ', j,BossList(j),RecMolecule(BossList(j))%Agg_Size
!!$     end do
!!$     stop
!!$  end if
!!$
!!$
!!$
  ! Check for reaction (collision)
```

```
 do k = 1,System_Info%Num_Particles ! loop over particles (receptor monomers)

  if (RecMolecule(k)%Boss == k) then ! this will only check for the bosses

      BossCount = BossCount+1  ! count bosses in this sub call


      AggRecCount = RecMolecule(k)%Agg_Size ! number of receptors on the aggregate

      Syk_Agg = pack(RecMolecule(:)%RecID, RecMolecule(:)%Boss==k)! holding the
recs in an aggregate

      !count each type of binding- each receptor- whether general syk binding is possible
through Iga, Igb
      if (System_Info%OutputLevel >=3) then
        WRITE(*,*) 'ending -- Syk_Agg has size ',size(Syk_Agg),'
AggRecCount=',AggRecCount
        do j=1,size(Syk_Agg)
          write(*,*) 'entry',j,':',Syk_Agg(j)
        end do
      endif


      ! initialize the prob vectors
      IgABindProb=0;
      IgBBindProb=0;

      ! to count eligible (at least one site unoccupied and phos>0) aggs
      ThisAggEligible=0
      ! count open (un-occupied) phos sites by phos level 1 or 2
      AggPhosCount=0

      do j = 1,AggRecCount ! this follows the receptors down the aggregate
        ! Iga binding requires three conditions.. no Lyn or Syk bound anf phos>0
        if (RecMolecule(Syk_Agg(j))%Iga_Syk == 0 .and.
RecMolecule(Syk_Agg(j))%Iga_Lyn == 0 .and. &
             RecMolecule(Syk_Agg(j))%Iga_Phos>0) then
          IgABindProb(j)=SykBindScaleFactor(RecMolecule(Syk_Agg(j))%Iga_Phos)

          ! count available sites by phos level (1 or 2)

AggPhosCount(RecMolecule(Syk_Agg(j))%Iga_Phos)=AggPhosCount(RecMolecule(Sy
k_Agg(j))%Iga_Phos)+1
          ThisAggEligible=1
        endif
        ! same for Igb
        if (RecMolecule(Syk_Agg(j))%Igb_Syk == 0 .and.
RecMolecule(Syk_Agg(j))%Igb_Lyn == 0 .and. &
           RecMolecule(Syk_Agg(j))%Igb_Phos>0) then
          IgBBindProb(j)=SykBindScaleFactor(2+RecMolecule(Syk_Agg(j))%Igb_Phos)

          ! count available sites by phos level (1 or 2)

AggPhosCount(RecMolecule(Syk_Agg(j))%Igb_Phos)=AggPhosCount(RecMolecule(Sy
k_Agg(j))%Igb_Phos)+1
          ThisAggEligible=1

        endif
      end do ! loop through recs in this agg

      ! counting eligible aggs in the system
      EligibleAggCount = EligibleAggCount + ThisAggEligible
      SysPhosCount = SysPhosCount + AggPhosCount

      distsq=&
         (RecMolecule(k)%Position(1)-SykMolecule(iSyk)%Position(1))**2+&
         (RecMolecule(k)%Position(2)-SykMolecule(iSyk)%Position(2))**2+&
         (RecMolecule(k)%Position(3)-SykMolecule(iSyk)%Position(3))**2

      if(distsq < MinDistNow .and.
sum(IgABindProb(1:AggRecCount))+sum(IgBBindProb(1:AggRecCount))>0) &
         MinDistNow=distsq /
(sum(IgABindProb(1:AggRecCount))+sum(IgBBindProb(1:AggRecCount)))

      ! compare the sq distance with the BR^2 scaled by the factors
      if(distsq <=
(sum(IgABindProb(1:AggRecCount))+sum(IgBBindProb(1:AggRecCount)))*Syk_Bind
Rad_Dimer**2 ) then
        ! ** implement the binding **

        !choose the site (rec+itam) according to the relative probs
        ! build a vector of probs
        SykBindProbVec(1:AggRecCount)=IgABindProb(1:AggRecCount)

SykBindProbVec(1+AggRecCount:2*AggRecCount)=IgBBindProb(1:AggRecCount)
        SykBindProbVec = SykBindProbVec /
sum(SykBindProbVec(1:2*AggRecCount)) ! normalize

        ! pick a kosher random number (guaranteed in [0,1])
        rannum=grnd()
        ! unfortunately this is necessary
        do while(rannum > 1.0 .and. rannum < 0.0)
          rannum = grnd()
        enddo

        CumSum=SykBindProbVec(1);
        j=1
        do while(CumSum < rannum .and. j <= 2*AggRecCount)
          j = j + 1
          CumSum = CumSum + SykBindProbVec(j)
        end do

        if (j <= AggRecCount) then
          ! ** Syk binds to Iga on Syk_Agg(j) **
          ! update Syk molecule
          SykMolecule(iSyk)%Receptor_ID = Syk_Agg(j)
          SykMolecule(iSyk)%Position = RecMolecule(Syk_Agg(j))%Position
          SykMolecule(iSyk)%Itam_site = 1
          SykMolecule(iSyk)%Syk_site = RecMolecule(Syk_Agg(j))%Iga_Phos ! binding
state of Syk given by phos level of dock site
          ! update receptor
          RecMolecule(Syk_Agg(j))%Iga_Syk = iSyk ! record Id of Syk molecule
        elseif (j <= 2*AggRecCount) then
          ! ** Syk binds to Igb on Syk_Agg(j - AggRecCount) **
          j = j - AggRecCount ! rec location on the chain
          ! update Syk molecule
          SykMolecule(iSyk)%Receptor_ID = Syk_Agg(j)
          SykMolecule(iSyk)%Position = RecMolecule(Syk_Agg(j))%Position
          SykMolecule(iSyk)%Itam_site = 2
          SykMolecule(iSyk)%Syk_site = RecMolecule(Syk_Agg(j))%Igb_Phos ! binding
state of Syk given by phos level of dock site
          ! update receptor
          RecMolecule(Syk_Agg(j))%Igb_Syk = iSyk ! record Id of Syk molecule
        endif

        System_Info%Free_Syk_Count = System_Info%Free_Syk_Count - 1

        if (System_Info%OutputLevel >=1) &
           write(*,FMT="('T=',f14.8,' SykBind SykID=',i4,' RecID=',i3,'.',i1,' Agg=',i3,'
Sz=',i0,' Free Syk ',i0)") &
           System_Info%Current_Simulation_Time,iSyk,
SykMolecule(iSyk)%Receptor_ID ,SykMolecule(iSyk)%Syk_site,&
           RecMolecule(Syk_Agg(j))%Boss,&
           RecMolecule(Syk_Agg(j))%Agg_Size,System_Info%Free_Syk_Count

        System_Info%Reaction = .true.
```

```
      endif ! if distance is less than scaled BR^2

    end if ! only check if receptor is a bossgregate
    ! TODO: randomize the order receptors are picked

    if (System_Info%Reaction) exit ! break the loop / so only one binding reaction is
possible

  end do ! loop over all receptors

  if(System_Info%Reaction) then
    Syk_Bind_Reaction =  Syk_Bind_Reaction + 1
  else
    Syk_Bind_NoReaction =  Syk_Bind_NoReaction + 1 ! calls that didn't lead to a
reaction
    ! report aggs and phos sites in the system but missed
    Enc_BossCount=Enc_BossCount+BossCount ! aggregates in the system  (add for
each call)
    Enc_EligAggCount = Enc_EligAggCount + EligibleAggCount ! eligible aggregates
by call
    Enc_SysPhosCount = Enc_SysPhosCount + SysPhosCount ! ph=1 and 2 phos sites in
the system this time (add for each call)
    if(MinDistNow < SysMinDist) SysMinDist=MinDistNow
  endif



  if (System_Info%OutputLevel >=2) write(*,*) 'SykBind end'


END SUBROUTINE SykBind

!******************************************************************************
*******************************************************************************
*************

SUBROUTINE SykUnbindReaction(iSyk)

  USE ParticleInfo
  USE ModelConstants
  USE mtmod

  IMPLICIT NONE

  INTEGER, INTENT(IN)::iSyk
  DOUBLE PRECISION :: placeangle_one,placeangle_two, x1, y1, z1, rannum
  INTEGER :: Receptor_bound_syk, Rec_Site

  ! receptor lyn is bound to
  ! SykMolecule(iSyk)%Receptor_ID = Receptor_bound_syk
  Receptor_bound_syk = SykMolecule(iSyk)%Receptor_ID
  Rec_Site = SykMolecule(iSyk)%Itam_site

 ! make sure the current position is set to that of the [boss of] the binding receptor
  SykMolecule(iSyk)%Position =
RecMolecule(RecMolecule(Receptor_bound_syk)%Boss)%Position

  if (System_Info%OutputLevel >=2) &
      write(*,*) 'SykUnBind begin -- SykID ',iSyk, ' RecID', Receptor_bound_syk


  placeangle_one = 2*Pi*grnd() ! full circle, in the xy plane
  placeangle_two = (Pi/2)*grnd() ! angle from vertical to the direction of the velocity

  x1 = sin(placeangle_two)*cos(placeangle_one)*Syk_UnbindRad_RestDimer
  y1 = sin(placeangle_two)*sin(placeangle_one)*Syk_UnbindRad_RestDimer
  z1 = cos(placeangle_two)*Syk_UnbindRad_RestDimer


  !this will update Syk position
  CALL PeriodicBC3(&
      SykMolecule(iSyk)%Position(1)+x1,SykMolecule(iSyk)%Position(2)+y1, &
      SykMolecule(iSyk)%Position(3)+z1, SykMolecule(iSyk)%Position(1), &
      SykMolecule(iSyk)%Position(2),SykMolecule(iSyk)%Position(3))

  ! Update Receptor first
  If (SykMolecule(iSyk)%Itam_site==1) then
    RecMolecule(Receptor_bound_syk)%Iga_Syk = 0
  else if (SykMolecule(iSyk)%Itam_site==2) then
    RecMolecule(Receptor_bound_syk)%Igb_syk = 0
  End if

  ! update Syk now
  SykMolecule(iSyk)%Receptor_ID = 0
  System_Info%Free_Syk_Count = System_Info%Free_Syk_Count + 1
  SykMolecule(iSyk)%Itam_site = 0
  SykMolecule(iSyk)%Syk_site = 0


  if (System_Info%OutputLevel >=1) &
```

```
     write(*,FMT="('T=',f14.8,' SykUnBi SykID=',i3,' RecID=',i3,'.',i1,' Agg=',i3,' Sz=',i0'
Free Syk ',i0)") &
     System_Info%Current_Simulation_Time,iSyk, Receptor_bound_syk,Rec_Site,&
     RecMolecule( Receptor_bound_syk)%Boss, RecMolecule(
Receptor_bound_syk)%Agg_Size,&
     System_Info%Free_Syk_Count

  if (System_Info%OutputLevel >=2) &
     write(*,*) 'SykUnBind end -- SykID ',iSyk, ' RecID', Receptor_bound_syk

END SUBROUTINE SykUnbindReaction
```

REFERENCES

Editorial (no authors) (2010). Hunting down heterogeneity. *Nat Chem Biol* 6(10), 691-691.

Ackermann, M. (2015). A functional perspective on phenotypic heterogeneity in microorganisms. *Nat Rev Micro* 13(8), 497-508. doi: 10.1038/nrmicro3491.

Alimandi, M., Romano, A., Curia, M.C., Muraro, R., Fedi, P., Aaronson, S.A., et al. (1995). Cooperative signaling of ErbB3 and ErbB2 in neoplastic transformation and human mammary carcinomas. *Oncogene* 10(9), 1813-1821.

Allison, K.H., and Sledge, G.W. (2014). Heterogeneity and cancer. *Oncology (Williston Park)* 28(9), 772-778.

Alt, F.W., Yancopoulos, G.D., Blackwell, T.K., Wood, C., Thomas, E., Boss, M., et al. (1984). Ordered rearrangement of immunoglobulin heavy chain variable region segments. *The EMBO Journal* 3(6), 1209-1219.

Altschuler, S.J., and Wu, L.F. (2010). Cellular Heterogeneity: Do Differences Make a Difference? *Cell* 141(4), 559-563. doi: 10.1016/j.cell.2010.04.033.

Ander, M., Beltrao, P., Di Ventura, B., Ferkinghoff-Borg, J., Foglierini, M., Kaplan, A., et al. (2004). SmartCell, a framework to simulate cellular processes that combines stochastic approximation with diffusion and localisation: analysis of simple networks. *Syst Biol (Stevenage)* 1(1), 129-138.

Andrews, N.L., Pfeiffer, J.R., Martinez, A.M., Haaland, D.M., Davis, R.W., Kawakami, T., et al. (2009a). Small, Mobile FcεRI Receptor Aggregates Are Signaling Competent. *Immunity* 31(3), 469-479. doi: https://doi.org/10.1016/j.immuni.2009.06.026.

Andrews, S.S., Addy, N.J., Brent, R., and Arkin, A.P. (2010). Detailed Simulations of Cell Biology with Smoldyn 2.1. *PLoS Comput Biol* 6(3). doi: 10.1371/journal.pcbi.1000705.

Andrews, S.S., and Bray, D. (2004). Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Phys Biol* 1(3-4), 137-151. doi: 10.1088/1478-3967/1/3/001.

Andrews, S.S., Dinh, T., and Arkin, A.P. (2009b). "Stochastic Models of Biological Processes," in *Encyclopedia of Complexity and Systems Science,* ed. R.A. Meyers. (New York, NY: Springer New York), 8730-8749.

Arjunan, S.N., and Tomita, M. (2010). A new multicompartmental reaction-diffusion modeling method links transient membrane attachment of E. coli MinE to E-ring formation. *Syst Synth Biol* 4(1)**,** 35-53. doi: 10.1007/s11693-009-9047-2.

Au, J.L.S., Jang, S.H., Zheng, J., Chen, C.T., Song, S., Hu, L., et al. (2001). Determinants of drug delivery and transport to solid tumors. *Journal of Controlled Release* 74(1–3)**,** 31-46. doi: http://dx.doi.org/10.1016/S0168-3659(01)00308-X.

Balkwill, F.R., Capasso, M., and Hagemann, T. (2012). The tumor microenvironment at a glance. *Journal of Cell Science* 125(23)**,** 5591-5596. doi: 10.1242/jcs.116392.

Bankovich, A.J., Raunser, S., Juo, Z.S., Walz, T., Davis, M.M., and Garcia, K.C. (2007). Structural insight into pre-B cell receptor function. *Science* 316(5822)**,** 291-294. doi: 10.1126/science.1139412.

Barua, D., Hlavacek, W.S., and Lipniacki, T. (2012). A computational model for early events in B cell antigen receptor signaling: analysis of the roles of Lyn and Fyn. *J Immunol* 189(2)**,** 646-658. doi: 10.4049/jimmunol.1102003.

Baselga, J., and Swain, S.M. (2009). Novel anticancer targets: revisiting ERBB2 and discovering ERBB3. *Nat Rev Cancer* 9(7)**,** 463-475. doi: 10.1038/nrc2656.

Baxter, L.T., and Jain, R.K. (1989). Transport of fluid and macromolecules in tumors. I. Role of interstitial pressure and convection. *Microvasc Res* 37(1)**,** 77-104.

Benschop, R.J., and Cambier, J.C. (1999). B cell development: signal transduction by antigen receptors and their surrogates. *Current Opinion in Immunology* 11(2)**,** 143-151. doi: http://dx.doi.org/10.1016/S0952-7915(99)80025-9.

Bicocca, Vincent T., Chang, Bill H., Masouleh, Behzad K., Muschen, M., Loriaux, Marc M., Druker, Brian J., et al. (2012). Crosstalk between ROR1 and the Pre-B Cell Receptor Promotes Survival of t(1;19) Acute Lymphoblastic Leukemia. *Cancer Cell* 22(5)**,** 656-667. doi: https://doi.org/10.1016/j.ccr.2012.08.027.

Boggon, T.J., and Eck, M.J. (2004). Structure and regulation of Src family kinases. *Oncogene* 23(48)**,** 7918-7927. doi: 10.1038/sj.onc.1208081.

Boulianne, L., Al Assaad, S., Dumontier, M., and Gross, W.J. (2008). GridCell: a stochastic particle-based biological system simulator. *BMC Systems Biology* 2**,** 66-66. doi: 10.1186/1752-0509-2-66.

Bunnell, S.C., Hong, D.I., Kardon, J.R., Yamazaki, T., McGlade, C.J., Barr, V.A., et al. (2002). T cell receptor ligation induces the formation of dynamically regulated signaling assemblies. *J Cell Biol* 158(7)**,** 1263-1275.

Burden, S., and Yarden, Y. (1997). Neuregulins and Their Receptors: A Versatile Signaling Module in Organogenesis and Oncogenesis. *Neuron* 18(6)**,** 847-855. doi: https://doi.org/10.1016/S0896-6273(00)80324-4.

Carlson, R.W., Allred, D.C., Anderson, B.O., Burstein, H.J., Carter, W.B., Edge, S.B., et al. (2009). Breast Cancer. *Journal of the National Comprehensive Cancer Network* 7(2)**,** 122-192. doi: 10.6004/jnccn.2009.0012.

Chang, A.Y.n., and Marshall, W.F.n. (2017). Organelles – understanding noise and heterogeneity in cell biology at an intermediate scale. *Journal Name: Journal of Cell Science; Journal Volume: 130; Journal Issue: 5; Conference: null; Patent File Date: null; Patent Priority Date: null; Other Information: null; Related Information: null***,** Medium: X; Size: 819 to 826; Quantity: null; OS: null; Compatibility: null; Other: null.

Chausovsky, A., Waterman, H., Elbaum, M., Yarden, Y., Geiger, B., and Bershadsky, A.D. (2000). Molecular requirements for the effect of neuregulin on cell spreading, motility and colony organization. *Oncogene* 19(7)**,** 878-888. doi: 10.1038/sj.onc.1203410.

Chen, Z., Shojaee, S., Buchner, M., Geng, H., Lee, J.W., Klemm, L., et al. (2015). Signalling thresholds and negative B-cell selection in acute lymphoblastic leukaemia. *Nature* 521(7552)**,** 357-361. doi: 10.1038/nature14231 http://www.nature.com/nature/journal/v521/n7552/abs/nature14231.html - supplementary-information.

Chung, I., Akita, R., Vandlen, R., Toomre, D., Schlessinger, J., and Mellman, I. (2010). Spatial control of EGF receptor activation by reversible dimerization on living cells. *Nature* 464(7289)**,** 783-787. doi: http://www.nature.com/nature/journal/v464/n7289/suppinfo/nature08827_S1.html.

Cianfrocca, M., and Goldstein, L.J. (2004). Prognostic and predictive factors in early-stage breast cancer. *Oncologist* 9(6)**,** 606-616. doi: 10.1634/theoncologist.9-6-606.

Cornall, R.J., Cheng, A.M., Pawson, T., and Goodnow, C.C. (2000). Role of Syk in B-cell development and antigen-receptor signaling. *Proceedings of the National Academy of Sciences of the United States of America* 97(4)**,** 1713-1718.

Costa, M.N., Radhakrishnan, K., and Edwards, J.S. (2011). Monte Carlo simulations of plasma membrane corral-induced EGFR clustering. *J Biotechnol* 151(3)**,** 261-270. doi: 10.1016/j.jbiotec.2010.12.009.

Costa, M.N., Radhakrishnan, K., Wilson, B.S., Vlachos, D.G., and Edwards, J.S. (2009). Coupled stochastic spatial and non-spatial simulations of ErbB1 signaling pathways demonstrate the importance of spatial organization in signal transduction. *PLoS One* 4(7)**,** e6316. doi: 10.1371/journal.pone.0006316.

Cui, Y., Zhang, X.P., Sun, Y.S., Tang, L., and Shen, L. (2008). Apparent diffusion coefficient: potential imaging biomarker for prediction and early detection of response to chemotherapy in hepatic metastases. *Radiology* 248(3)**,** 894-900.

Curti, B.D., Urba, W.J., Alvord, W.G., Janik, J.E., Smith, J.W., 2nd, Madara, K., et al. (1993). Interstitial pressure of subcutaneous nodules in melanoma and lymphoma patients: changes during treatment. *Cancer Res* 53(10 Suppl)**,** 2204-2207.

Day, C.A., and Kenworthy, A.K. (2009). Tracking microdomain dynamics in cell membranes. *Biochimica et Biophysica Acta (BBA) - Biomembranes* 1788(1)**,** 245-253. doi: https://doi.org/10.1016/j.bbamem.2008.10.024.

Devore, J.L. (2011). *Probability and Statistics for Engineering and the Sciences.* Brooks/Cole, Cengage Learning.

Dinic, J., Riehl, A., Adler, J., and Parmryd, I. (2015). The T cell receptor resides in ordered plasma membrane nanodomains that aggregate upon patching of the receptor. *Scientific Reports* 5**,** 10082. doi: 10.1038/srep10082
https://www.nature.com/articles/srep10082 - supplementary-information.

Dose Schwarz, J., Bader, M., Jenicke, L., Hemminger, G., Janicke, F., and Avril, N. (2005). Early prediction of response to chemotherapy in metastatic breast cancer using sequential 18F-FDG PET. *J Nucl Med* 46(7)**,** 1144-1150.

Douglass, A.D., and Vale, R.D. (2005). Single-molecule microscopy reveals plasma membrane microdomains created by protein-protein networks that exclude or trap signaling molecules in T cells. *Cell* 121(6)**,** 937-950. doi: 10.1016/j.cell.2005.04.009.

Duy, C., Yu, J.J., Nahar, R., Swaminathan, S., Kweon, S.-M., Polo, J.M., et al. (2010). BCL6 is critical for the development of a diverse primary B cell repertoire. *The Journal of Experimental Medicine* 207(6)**,** 1209-1221. doi: 10.1084/jem.20091299.

Edwards, M.S., Chadda, S.D., Zhao, Z., Barber, B.L., and Sykes, D.P. (2012). A systematic review of treatment guidelines for metastatic colorectal cancer. *Colorectal Dis* 14(2)**,** e31-47. doi: 10.1111/j.1463-1318.2011.02765.x.

Egger, M.E., Cannon, R.M., Metzger, T.L., Nowacki, M., Kelly, L., Tatum, C., et al. (2013). Assessment of chemotherapy response in colorectal liver metastases in patients undergoing hepatic resection and the correlation to pathologic residual viable tumor. *J Am Coll Surg* 216(4)**,** 845-856.

Elowitz, M.B., Levine, A.J., Siggia, E.D., and Swain, P.S. (2002). Stochastic Gene Expression in a Single Cell. *Science* 297(5584)**,** 1183-1186. doi: 10.1126/science.1070919.

Erasmus, M.F., Matlawska-Wasowska, K., Kinjyo, I., Mahajan, A., Winter, S.S., Xu, L., et al. (2016). Dynamic pre-BCR homodimers fine-tune autonomous survival signals in B cell precursor acute lymphoblastic leukemia. *Science Signaling* 9(456)**,** ra116-ra116. doi: 10.1126/scisignal.aaf3949.

Erban, R. (2014). From molecular dynamics to Brownian dynamics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science* 470(2167). doi: 10.1098/rspa.2014.0036.

Faeder, J.R., Hlavacek, W.S., Reischl, I., Blinov, M.L., Metzger, H., Redondo, A., et al. (2003). Investigation of early events in Fc epsilon RI-mediated signaling using a detailed mathematical model. *J Immunol* 170(7)**,** 3769-3781.

Field, K.A., Holowka, D., and Baird, B. (1997). Compartmentalized Activation of the High Affinity Immunoglobulin E Receptor within Membrane Domains. *Journal of Biological Chemistry* 272(7)**,** 4276-4280. doi: 10.1074/jbc.272.7.4276.

Fleming, H.E., and Paige, C.J. (2001). Pre-B Cell Receptor Signaling Mediates Selective Response to IL-7 at the Pro-B to Pre-B Cell Transition via an ERK/MAP Kinase-

Dependent Pathway. *Immunity* 15(4)**,** 521-531. doi: http://dx.doi.org/10.1016/S1074-7613(01)00216-3.

Frieboes, H.B., Edgerton, M.E., Fruehauf, J.P., Rose, F.R.A.J., Worrall, L.K., Gatenby, R.A., et al. (2009). Prediction of drug response in breast cancer using integrative experimental/computational modeling. *Cancer research* 69(10)**,** 4484-4492. doi: 10.1158/0008-5472.CAN-08-3740.

Fuentes-Panana, E.M., Bannish, G., Shah, N., and Monroe, J.G. (2004). Basal Igalpha/Igbeta signals trigger the coordinated initiation of pre-B cell antigen receptor-dependent processes. *J Immunol* 173(2)**,** 1000-1011.

Gallinger, S., Biagi, J.J., Fletcher, G.G., Nhan, C., Ruo, L., and McLeod, R.S. (2013). Liver resection for colorectal cancer metastases. *Curr Oncol* 20(3).

Gassmann, M., Casagranda, F., Orioli, D., Simon, H., Lai, C., Klein, R., et al. (1995). Aberrant neural and cardiac development in mice lacking the ErbB4 neuregulin receptor. *Nature* 378(6555)**,** 390-394. doi: 10.1038/378390a0.

Gauld, S.B., and Cambier, J.C. (2004). Src-family kinases in B-cell development and signaling. *Oncogene* 23(48)**,** 8001-8006. doi: 10.1038/sj.onc.1208075.

Gaus, K., Chklovskaia, E., Fazekas de St. Groth, B., Jessup, W., and Harder, T. (2005). Condensation of the plasma membrane at the site of T lymphocyte activation. *The Journal of Cell Biology* 171(1)**,** 121-131. doi: 10.1083/jcb.200505047.

Gauthier, L., Rossi, B., Roux, F., Termine, E., and Schiff, C. (2002). Galectin-1 is a stromal cell ligand of the pre-B cell receptor (BCR) implicated in synapse formation between pre-B and stromal cells and in pre-BCR triggering. *Proceedings of the National Academy of Sciences* 99(20)**,** 13014-13019. doi: 10.1073/pnas.202323999.

Geier, J.K., and Schlissel, M.S. (2006). Pre-BCR signals and the control of Ig gene rearrangements. *Semin Immunol* 18(1)**,** 31-39. doi: 10.1016/j.smim.2005.11.001.

Geng, H., Hurtz, C., Lenz, K.B., Chen, Z., Baumjohann, D., Thompson, S., et al. (2015). Self-enforcing feedback activation between BCL6 and pre-B cell receptor signaling defines a distinct subtype of acute lymphoblastic leukemia. *Cancer Cell* 27(3)**,** 409-425. doi: 10.1016/j.ccell.2015.02.003.

Gillespie, D.T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics* 22(4)**,** 403-434. doi: http://dx.doi.org/10.1016/0021-9991(76)90041-3.

Gillespie, D.T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* 81(25)**,** 2340-2361. doi: 10.1021/j100540a008.

Glazer, E.S., Beaty, K., Abdalla, E.K., Vauthey, J.N., and Curley, S.A. (2010). Effectiveness of positron emission tomography for predicting chemotherapy response in colorectal cancer liver metastases. *Arch Surg* 145(4)**,** 340-345.

Goel, S., Duda, D.G., Xu, L., Munn, L.L., Boucher, Y., Fukumura, D., et al. (2011). NORMALIZATION OF THE VASCULATURE FOR TREATMENT OF CANCER AND OTHER DISEASES. *Physiological Reviews* 91(3)**,** 1071-1121. doi: 10.1152/physrev.00038.2010.

Goñi, F.M. (2014). The basic structure and dynamics of cell membranes: An update of the Singer–Nicolson model. *Biochimica et Biophysica Acta (BBA) - Biomembranes* 1838(6)**,** 1467-1476. doi: https://doi.org/10.1016/j.bbamem.2014.01.006.

Gottesman, M.M. (2002). Mechanisms of cancer drug resistance. *Annu Rev Med* 53**,** 615-627.

Grantab, R., Sivananthan, S., and Tannock, I.F. (2006). The Penetration of Anticancer Drugs through Tumor Tissue as a Function of Cellular Adhesion and Packing Density of Tumor Cells. *Cancer Research* 66(2)**,** 1033-1039. doi: 10.1158/0008-5472.can-05-3077.

Grantab, R., and Tannock, I. (2012). Penetration of anticancer drugs through tumour tissue as a function of cellular packing density and interstitial fluid pressure and its modification by bortezomib. *BMC Cancer* 12(1)**,** 214.

GraphPad Software (2007). "GraphPad Prism, Version 5.0, Statistics Guide". (http://www.graphpad.com/downloads/docs/Prism5Stats.pdf).

Grecco, Hernán E., Schmick, M., and Bastiaens, Philippe I.H. Signaling from the Living Plasma Membrane. *Cell* 144(6)**,** 897-909. doi: 10.1016/j.cell.2011.01.029.

Groves, J.T., and Kuriyan, J. (2010). Molecular mechanisms in signal transduction at the membrane. *Nat Struct Mol Biol* 17(6)**,** 659-665. doi: 10.1038/nsmb.1844.

Guo, B., Kato, R.M., Garcia-Lloret, M., Wahl, M.I., and Rawlings, D.J. (2000). Engagement of the Human Pre-B Cell Receptor Generates a Lipid Raft–Dependent Calcium Signaling Complex. *Immunity* 13(2)**,** 243-253. doi: https://doi.org/10.1016/S1074-7613(00)00024-8.

Guy, P.M., Platko, J.V., Cantley, L.C., Cerione, R.A., and Carraway, K.L., 3rd (1994). Insect cell-expressed p180erbB3 possesses an impaired tyrosine kinase activity. *Proc Natl Acad Sci U S A* 91(17)**,** 8132-8136.

Hanahan, D., and Weinberg, R.A. (2000). The Hallmarks of Cancer. *Cell* 100(1)**,** 57-70. doi: https://doi.org/10.1016/S0092-8674(00)81683-9.

Hanahan, D., and Weinberg, Robert A. (2011). Hallmarks of Cancer: The Next Generation. *Cell* 144(5)**,** 646-674. doi: https://doi.org/10.1016/j.cell.2011.02.013.

Harry, V.N., Semple, S.I., Parkin, D.E., and Gilbert, F.J. (2010). Use of new imaging techniques to predict tumour response to therapy. *The Lancet Oncology* 11(1)**,** 92-102. doi: http://dx.doi.org/10.1016/S1470-2045(09)70190-1.

Hashizume, H., Baluk, P., Morikawa, S., McLean, J.W., Thurston, G., Roberge, S., et al. (2000). Openings between Defective Endothelial Cells Explain Tumor Vessel Leakiness. *The American Journal of Pathology* 156(4)**,** 1363-1380.

Hattne, J., Fange, D., and Elf, J. (2005). Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics* 21(12)**,** 2923-2924. doi: 10.1093/bioinformatics/bti431.

Heldin, C.H., Rubin, K., Pietras, K., and Ostman, A. (2004). High interstitial fluid pressure - an obstacle in cancer therapy. *Nat Rev Cancer* 4(10)**,** 806-813.

Herzog, S., Reth, M., and Jumaa, H. (2009). Regulation of B-cell proliferation and differentiation by pre-B-cell receptor signalling. *Nat Rev Immunol* 9(3)**,** 195-205.

Hobbs, S.K., Monsky, W.L., Yuan, F., Roberts, W.G., Griffith, L., Torchilin, V.P., et al. (1998). Regulation of transport pathways in tumor vessels: role of tumor type and microenvironment. *Proc Natl Acad Sci U S A* 95(8)**,** 4607-4612.

Hoppe, A., and Low-Nam, S. (2014). "Live-Cell TIRF Imaging of Molecular Assembly and Plasma Membrane Topography," in *Cell Membrane Nanodomains*. CRC Press), 261-280.

Hsieh, M.Y., Yang, S., Raymond-Stinz, M.A., Edwards, J.S., and Wilson, B.S. (2010). Spatio-temporal modeling of signaling protein recruitment to EGFR. *BMC Syst Biol* 4**,** 57. doi: 10.1186/1752-0509-4-57.

Hsieh, M.Y., Yang, S., Raymond-Stinz, M.A., Steinberg, S., Vlachos, D.G., Shu, W., et al. (2008). Stochastic simulations of ErbB homo and heterodimerisation: potential impacts of receptor conformational state and spatial segregation. *IET Syst Biol* 2(5)**,** 256-272. doi: 10.1049/iet-syb:20070073.

Huang, S., Li, C., Armstrong, E.A., Peet, C.R., Saker, J., Amler, L.C., et al. (2013). Dual Targeting of EGFR and HER3 with MEHD7945A Overcomes Acquired Resistance to EGFR Inhibitors and Radiation. *Cancer Research* 73(2)**,** 824-833. doi: 10.1158/0008-5472.can-12-1611.

Hubbard, S.R., and Till, J.H. (2000). Protein tyrosine kinase structure and function. *Annu Rev Biochem* 69**,** 373-398. doi: 10.1146/annurev.biochem.69.1.373.

Huh, D., and Paulsson, J. (2011). Non-genetic heterogeneity from stochastic partitioning at cell division. *Nat Genet* 43(2)**,** 95-100.

Ingley, E. (2012). Functions of the Lyn tyrosine kinase in health and disease. *Cell Communication and Signaling* 10(1)**,** 21. doi: 10.1186/1478-811x-10-21.

Jain, R.K. (1987). Transport of molecules in the tumor interstitium: a review. *Cancer Res* 47(12)**,** 3039-3051.

Jain, R.K. (1989). Delivery of novel therapeutic agents in tumors: physiological barriers and strategies. *J Natl Cancer Inst* 81(8)**,** 570-576.

Jain, R.K. (1990). Physiological barriers to delivery of monoclonal antibodies and other macromolecules in tumors. *Cancer Res* 50(3 Suppl)**,** 814s-819s.

Jain, R.K. (2005). Normalization of tumor vasculature: an emerging concept in antiangiogenic therapy. *Science* 307(5706)**,** 58-62. doi: 10.1126/science.1104819.

Jang, S., Wientjes, M.G., Lu, D., and Au, J.S. (2003). Drug Delivery and Transport to Solid Tumors. *Pharmaceutical Research* 20(9)**,** 1337-1350. doi: 10.1023/a:1025785505977.

Johnson, S.A., Pleiman, C.M., Pao, L., Schneringer, J., Hippen, K., and Cambier, J.C. (1995). Phosphorylated immunoreceptor signaling motifs (ITAMs) exhibit unique abilities to bind and activate Lyn and Syk tyrosine kinases. *J Immunol* 155(10)**,** 4596-4603.

Junttila, M.R., and de Sauvage, F.J. (2013). Influence of tumour micro-environment heterogeneity on therapeutic response. *Nature* 501(7467)**,** 346-354. doi: 10.1038/nature12626.

Kaizuka, Y., Douglass, A.D., Varma, R., Dustin, M.L., and Vale, R.D. (2007). Mechanisms for segregating T cell receptor and adhesion molecules during immunological synapse formation in Jurkat T cells. *Proceedings of the National Academy of Sciences* 104(51)**,** 20296-20301. doi: 10.1073/pnas.0710258105.

Kanas, G.P., Taylor, A., Primrose, J.N., Langeberg, W.J., Kelsh, M.A., Mowat, F.S., et al. (2012). Survival after liver resection in metastatic colorectal cancer: review and meta-analysis of prognostic factors. *Clin Epidemiol* 4**,** 283-301.

Kerketta, R., Halasz, A.M., Steinkamp, M.P., Wilson, B.S., and Edwards, J.S. (2016). Effect of Spatial Inhomogeneities on the Membrane Surface on Receptor Dimerization and Signal Initiation. *Front Cell Dev Biol* 4**,** 81. doi: 10.3389/fcell.2016.00081.

Keshvara, L.M., Isaacson, C.C., Yankee, T.M., Sarac, R., Harrison, M.L., and Geahlen, R.L. (1998). Syk- and Lyn-dependent phosphorylation of Syk on multiple tyrosines following B cell activation includes a site that negatively regulates signaling. *J Immunol* 161(10)**,** 5276-5283.

Kim, M., Gillies, R.J., and Rejniak, K.A. (2013). Current Advances in Mathematical Modeling of Anti-Cancer Drug Penetration into Tumor Tissues. *Frontiers in Oncology* 3**,** 278. doi: 10.3389/fonc.2013.00278.

Koh, D.M., and Padhani, A.R. (2006). Diffusion-weighted MRI: a new functional clinical technique for tumour imaging. *Br J Radiol* 79(944)**,** 633-635.

Koh, D.M., Scurr, E., Collins, D., Kanber, B., Norman, A., Leach, M.O., et al. (2007). Predicting response of colorectal hepatic metastasis: value of pretreatment apparent diffusion coefficients. *AJR Am J Roentgenol* 188(4)**,** 1001-1008.

Kohler, F., Hug, E., Eschbach, C., Meixlsperger, S., Hobeika, E., Kofer, J., et al. (2008). Autoreactive B cell receptors mimic autonomous pre-B cell receptor signaling and induce proliferation of early B cells. *Immunity* 29(6), 912-921. doi: 10.1016/j.immuni.2008.10.013.

Kuh, H.J., Jang, S.H., Wientjes, M.G., Weaver, J.R., and Au, J.L. (1999). Determinants of paclitaxel penetration and accumulation in human solid tumor. *J Pharmacol Exp Ther* 290(2), 871-880.

Kurosaki, T., Johnson, S.A., Pao, L., Sada, K., Yamamura, H., and Cambier, J.C. (1995). Role of the Syk autophosphorylation site and SH2 domains in B cell antigen receptor signaling. *J Exp Med* 182(6), 1815-1823.

Kurosaki, T., Maeda, A., Ishiai, M., Hashimoto, A., Inabe, K., and Takata, M. (2000). Regulation of the phospholipase C-gamma2 pathway in B cells. *Immunol Rev* 176, 19-29.

Kusumi, A., Nakada, C., Ritchie, K., Murase, K., Suzuki, K., Murakoshi, H., et al. (2005). Paradigm shift of the plasma membrane concept from the two-dimensional continuum fluid to the partitioned fluid: high-speed single-molecule tracking of membrane molecules. *Annu Rev Biophys Biomol Struct* 34, 351-378. doi: 10.1146/annurev.biophys.34.040204.144637.

Kusumi, A., and Sako, Y. (1996). Cell surface organization by the membrane skeleton. *Curr Opin Cell Biol* 8(4), 566-574.

Kusumi, A., Sako, Y., and Yamamoto, M. (1993). Confined lateral diffusion of membrane receptors as studied by single particle tracking (nanovid microscopy). Effects of calcium-induced differentiation in cultured epithelial cells. *Biophys J* 65(5), 2021-2040. doi: 10.1016/s0006-3495(93)81253-0.

Kyle, A.H., Huxham, L.A., Chiam, A.S., Sim, D.H., and Minchinton, A.I. (2004). Direct assessment of drug penetration into tissue using a novel application of three-dimensional cell culture. *Cancer Res* 64(17), 6304-6309.

Kyle, A.H., Huxham, L.A., Yeoman, D.M., and Minchinton, A.I. (2007). Limited tissue penetration of taxanes: a mechanism for resistance in solid tumors. *Clin Cancer Res* 13(9), 2804-2810.

Lagerholm, B.C., Weinreb, G.E., Jacobson, K., and Thompson, N.L. (2005). Detecting microdomains in intact cell membranes. *Annu Rev Phys Chem* 56, 309-336. doi: 10.1146/annurev.physchem.56.092503.141211.

LeBien, T.W., and Tedder, T.F. (2008). B lymphocytes: how they develop and function. *Blood* 112(5)**,** 1570-1580. doi: 10.1182/blood-2008-02-078071.

Lee, K.F., Simon, H., Chen, H., Bates, B., Hung, M.C., and Hauser, C. (1995). Requirement for neuregulin receptor erbB2 in neural and cardiac development. *Nature* 378(6555)**,** 394-398. doi: 10.1038/378394a0.

Lee, Y., Ma, J., Lyu, H., Huang, J., Kim, A., and Liu, B. (2014). Role of erbB3 receptors in cancer therapeutic resistance. *Acta Biochim Biophys Sin (Shanghai)* 46(3)**,** 190-198. doi: 10.1093/abbs/gmt150.

Leonard, G.D., Brenner, B., and Kemeny, N.E. (2005). Neoadjuvant chemotherapy before liver resection for patients with unresectable liver metastases from colorectal carcinoma. *J Clin Oncol* 23(9)**,** 2038-2048.

Li, C., Krishnan, J., Stebbing, J., and Xu, X.Y. (2011). Use of mathematical models to understand anticancer drug delivery and its effect on solid tumors. *Pharmacogenomics* 12(9)**,** 1337-1348.

Lidke, D.S., and Wilson, B.S. (2009). Caught in the Act: Quantifying Protein Behavior in Living Cells. *Trends in cell biology* 19(11)**,** 566-574. doi: 10.1016/j.tcb.2009.08.004.

Lillemeier, B.F., Pfeiffer, J.R., Surviladze, Z., Wilson, B.S., and Davis, M.M. (2006). Plasma membrane-associated proteins are clustered into islands attached to the cytoskeleton. *Proceedings of the National Academy of Sciences* 103(50)**,** 18992-18997. doi: 10.1073/pnas.0609009103.

Liu, X., Hwang, H., Cao, L., Buckland, M., Cunningham, A., Chen, J., et al. (1998). Domain-specific gene disruption reveals critical regulation of neuregulin signaling by its cytoplasmic tail. *Proceedings of the National Academy of Sciences* 95(22)**,** 13024-13029. doi: 10.1073/pnas.95.22.13024.

Lorenzo, G., Scott, M.A., Tew, K., Hughes, T.J.R., Zhang, Y.J., Liu, L., et al. (2016). Tissue-scale, personalized modeling and simulation of prostate cancer growth. *Proceedings of the National Academy of Sciences* 113(48)**,** E7663-E7671. doi: 10.1073/pnas.1615791113.

Marguet, D., Lenne, P.-F., Rigneault, H., and He, H.-T. (2006). Dynamics in the plasma membrane: how to combine fluidity and order. *The EMBO Journal* 25(15)**,** 3446-3457. doi: 10.1038/sj.emboj.7601204.

Marshall, W.F. (2012). Organelle Size Control Systems: From Cell Geometry to Organelle-Directed Medicine. *BioEssays : news and reviews in molecular, cellular and developmental biology* 34(9)**,** 721-724. doi: 10.1002/bies.201200043.

Mårtensson, I.-L., Almqvist, N., Grimsholm, O., and Bernardi, A.I. (2010). The pre-B cell receptor checkpoint. *FEBS Letters* 584(12)**,** 2572-2579. doi: http://dx.doi.org/10.1016/j.febslet.2010.04.057.

Mayawala, K., Vlachos, D.G., and Edwards, J.S. (2005a). Computational modeling reveals molecular details of epidermal growth factor binding. *BMC Cell Biol* 6**,** 41. doi: 10.1186/1471-2121-6-41.

Mayawala, K., Vlachos, D.G., and Edwards, J.S. (2005b). Heterogeneities in EGF receptor density at the cell surface can lead to concave up scatchard plot of EGF binding. *FEBS Lett* 579(14)**,** 3043-3047. doi: 10.1016/j.febslet.2005.04.059.

Mayawala, K., Vlachos, D.G., and Edwards, J.S. (2006). Spatial modeling of dimerization reaction dynamics in the plasma membrane: Monte Carlo vs. continuum differential equations. *Biophys Chem* 121(3)**,** 194-208. doi: 10.1016/j.bpc.2006.01.008.

Meacham, C.E., and Morrison, S.J. (2013). Tumor heterogeneity and cancer cell plasticity. *Nature* 501(7467)**,** 328-337. doi: 10.1038/nature12624.

Meffre, E., Casellas, R., and Nussenzweig, M.C. (2000). Antibody regulation of B cell development. *Nat Immunol* 1(5)**,** 379-385.

Meyer, D., and Birchmeier, C. (1995). Multiple essential functions of neuregulin in development. *Nature* 378(6555)**,** 386-390. doi: 10.1038/378386a0.

Miettinen, P.J., Berger, J.E., Meneses, J., Phung, Y., Pedersen, R.A., Werb, Z., et al. (1995). Epithelial immaturity and multiorgan failure in mice lacking epidermal growth factor receptor. *Nature* 376(6538)**,** 337-341.

Milosevic, M., Fyles, A., Hedley, D., Pintilie, M., Levin, W., Manchul, L., et al. (2001). Interstitial fluid pressure predicts survival in patients with cervix cancer independent of clinical prognostic factors and tumor oxygen measurements. *Cancer Res* 61(17)**,** 6400-6405.

Minchinton, A.I., and Tannock, I.F. (2006). Drug penetration in solid tumours. *Nat Rev Cancer* 6(8)**,** 583-592. doi: 10.1038/nrc1893.

Monroe, J.G. (2006). ITAM-mediated tonic signalling through pre-BCR and BCR complexes. *Nat Rev Immunol* 6(4)**,** 283-294.

Mukherjee, S., Zhu, J., Zikherman, J., Parameswaran, R., Kadlecek, T.A., Wang, Q., et al. (2013). Monovalent and Multivalent Ligation of the B Cell Receptor Exhibit Differential Dependence upon Syk and Src Family Kinases. *Science Signaling* 6(256)**,** ra1-ra1. doi: 10.1126/scisignal.2003220.

Muraca, M. (1994). *Methods in Biliary Research.* United Stated of America: CRC Press.

Müschen, M. (2015). Rationale for targeting the pre–B-cell receptor signaling pathway in acute lymphoblastic leukemia. *Blood* 125(24)**,** 3688-3693. doi: 10.1182/blood-2015-01-567842.

Nagy, J.A., Chang, S.H., Dvorak, A.M., and Dvorak, H.F. (2009). Why are tumour blood vessels abnormal and why is it important to know? *British Journal of Cancer* 100(6)**,** 865-869. doi: 10.1038/sj.bjc.6604929.

Nagy, P., Vereb, G., Sebestyén, Z., Horváth, G., Lockett, S.J., Damjanovich, S., et al. (2002). Lipid rafts and the local density of ErbB proteins influence the biological role of homo- and heteroassociations of ErbB2. *Journal of Cell Science* 115(22)**,** 4251-4262. doi: 10.1242/jcs.00118.

Nelson, C.M., VanDuijn, M.M., Inman, J.L., Fletcher, D.A., and Bissell, M.J. (2006). Tissue Geometry Determines Sites of Mammary Branching Morphogenesis in Organotypic Cultures. *Science* 314(5797)**,** 298-300. doi: 10.1126/science.1131000.

Ohnishi, K., and Melchers, F. (2003). The nonimmunoglobulin portion of [lambda]5 mediates cell-autonomous pre-B cell receptor signaling. *Nat Immunol* 4(9)**,** 849-856.

Organization, W.H. (1979). *WHO handbook for reporting results of cancer treatment* Geneva : World Health Organization.

Owen, D.M., Williamson, D., Rentero, C., and Gaus, K. (2009). Quantitative microscopy: protein dynamics and membrane organisation. *Traffic* 10(8)**,** 962-971. doi: 10.1111/j.1600-0854.2009.00908.x.

Ozbudak, E.M., Thattai, M., Kurtser, I., Grossman, A.D., and van Oudenaarden, A. (2002). Regulation of noise in the expression of a single gene. *Nat Genet* 31(1)**,** 69-73.

Pao, L.I., Famiglietti, S.J., and Cambier, J.C. (1998). Asymmetrical phosphorylation and function of immunoreceptor tyrosine-based activation motif tyrosines in B cell antigen receptor signal transduction. *J Immunol* 160(7)**,** 3305-3314.

Parsons, S.J., and Parsons, J.T. (2004). Src family kinases, key regulators of signal transduction. *Oncogene* 23(48)**,** 7906-7909. doi: 10.1038/sj.onc.1208160.

Pascal, J., Ashley, C.E., Wang, Z., Brocato, T.A., Butner, J.D., Carnes, E.C., et al. (2013a). Mechanistic modeling identifies drug-uptake history as predictor of tumor drug resistance and nano-carrier-mediated response. *ACS Nano* 7(12)**,** 11174-11182. doi: 10.1021/nn4048974.

Pascal, J., Bearer, E.L., Wang, Z., Koay, E.J., Curley, S.A., and Cristini, V. (2013b). Mechanistic patient-specific predictive correlation of tumor drug response with microenvironment and perfusion measurements. *Proc Natl Acad Sci U S A* 110(35)**,** 14266-14271.

Pike, L.J. (2003). Lipid rafts: bringing order to chaos. *J Lipid Res* 44(4)**,** 655-667. doi: 10.1194/jlr.R200021-JLR200.

Pinkas-Kramarski, R., Soussan, L., Waterman, H., Levkowitz, G., Alroy, I., Klapper, L., et al. (1996). Diversification of Neu differentiation factor and epidermal growth factor signaling by combinatorial receptor interactions. *The EMBO Journal* 15(10)**,** 2452-2467.

Pleiman, C.M., Abrams, C., Gauen, L.T., Bedzyk, W., Jongstra, J., Shaw, A.S., et al. (1994). Distinct p53/56lyn and p59fyn domains associate with nonphosphorylated and phosphorylated Ig-alpha. *Proc Natl Acad Sci U S A* 91(10)**,** 4268-4272.

Primeau, A.J., Rendon, A., Hedley, D., Lilge, L., and Tannock, I.F. (2005). The distribution of the anticancer drug Doxorubicin in relation to blood vessels in solid tumors. *Clin Cancer Res* 11(24 Pt 1)**,** 8782-8788.

Pryor, Meghan M., Low-Nam, Shalini T., Halász, Ádám M., Lidke, Diane S., Wilson, Bridget S., and Edwards, Jeremy S. (2013). Dynamic Transition States of ErbB1 Phosphorylation Predicted by Spatial Stochastic Modeling. *Biophysical Journal* 105(6)**,** 1533-1543. doi: 10.1016/j.bpj.2013.07.056.

Pryor, M.M., Steinkamp, M.P., Halasz, A.M., Chen, Y., Yang, S., Smith, M.S., et al. (2015). Orchestration of ErbB3 signaling through heterointeractions and homointeractions. *Molecular biology of the cell* 26(22)**,** 4109-4123.

Radhakrishnan, K., Halász, Á., McCabe, M.M., Edwards, J.S., and Wilson, B.S. (2012). Mathematical Simulation of Membrane Protein Clustering for Efficient Signal Transduction. *Annals of Biomedical Engineering* 40(11)**,** 2307-2318. doi: 10.1007/s10439-012-0599-z.

Rajewsky, K. (1996). Clonal selection and learning in the antibody system. *Nature* 381(6585)**,** 751-758.

Rao, A., Luo, C., and Hogan, P.G. (1997). Transcription factors of the NFAT family: regulation and function. *Annu Rev Immunol* 15**,** 707-747. doi: 10.1146/annurev.immunol.15.1.707.

Rejniak, K.A., Estrella, V., Chen, T., Cohen, A.S., Lloyd, M.C., and Morse, D.L. (2013). The role of tumor tissue architecture in treatment penetration and efficacy: an integrative study. *Frontiers in oncology* 3.

Rickert, R.C. (2013). New insights into pre-BCR and BCR signalling with relevance to B cell malignancies. *Nat Rev Immunol* 13(8)**,** 578-591. doi: 10.1038/nri3487.

Rodriguez, J.V., Kaandorp, J.A., Dobrzynski, M., and Blom, J.G. (2006). Spatial stochastic modelling of the phosphoenolpyruvate-dependent phosphotransferase (PTS) pathway in Escherichia coli. *Bioinformatics* 22(15)**,** 1895-1901. doi: 10.1093/bioinformatics/btl271.

Roskoski, R., Jr. (2014). The ErbB/HER family of protein-tyrosine kinases and cancer. *Pharmacol Res* 79**,** 34-74. doi: 10.1016/j.phrs.2013.11.002.

Ross, J.S., and Fletcher, J.A. (1998). The HER-2/neu Oncogene in Breast Cancer: Prognostic Factor, Predictive Factor, and Target for Therapy. *STEM CELLS* 16(6)**,** 413-428. doi: 10.1002/stem.160413.

Rubbia-Brandt, L., Giostra, E., Brezault, C., Roth, A.D., Andres, A., Audard, V., et al. (2007). Importance of histological tumor response assessment in predicting the outcome in patients with colorectal liver metastases treated with neo-adjuvant chemotherapy followed by liver surgery. *Ann Oncol* 18(2)**,** 299-304.

Rubin E, R.H. (2009). *Essentials of Rubin's Pathology.* Philadelphia: Lippincott Williams & Wilkins.

Sanga, S., Sinek, J.P., Frieboes, H.B., Ferrari, M., Fruehauf, J.P., and Cristini, V. (2006). Mathematical modeling of cancer progression and response to chemotherapy. *Expert Rev Anticancer Ther* 6(10)**,** 1361-1376.

Sato, Y., Yashiro, M., and Takakura, N. (2013). Heregulin induces resistance to lapatinib-mediated growth inhibition of HER2-amplified cancer cells. *Cancer Science* 104(12)**,** 1618-1625. doi: 10.1111/cas.12290.

Schirin-Sokhan, R., Winograd, R., Roderburg, C., Bubenzer, J., do, O.N., Guggenberger, D., et al. (2012). Response evaluation of chemotherapy in metastatic colorectal cancer by contrast enhanced ultrasound. *World J Gastroenterol* 18(6)**,** 541-545.

Sergina, N.V., Rausch, M., Wang, D., Blair, J., Hann, B., Shokat, K.M., et al. (2007). Escape from HER-family tyrosine kinase inhibitor therapy by the kinase-inactive HER3. *Nature* 445(7126)**,** 437-441. doi: http://www.nature.com/nature/journal/v445/n7126/suppinfo/nature05474_S1.html.

Shi, F., Telesco, S.E., Liu, Y., Radhakrishnan, R., and Lemmon, M.A. (2010). ErbB3/HER3 intracellular domain is competent to bind ATP and catalyze autophosphorylation. *Proceedings of the National Academy of Sciences* 107(17)**,** 7692-7697. doi: 10.1073/pnas.1002753107.

Sibilia, M., Steinbach, J.P., Stingl, L., Aguzzi, A., and Wagner, E.F. (1998). A strain-independent postnatal neurodegeneration in mice lacking the EGF receptor. *The EMBO Journal* 17(3)**,** 719-731. doi: 10.1093/emboj/17.3.719.

Siegel, R., Desantis, C., and Jemal, A. (2014). Colorectal cancer statistics, 2014. *CA Cancer J Clin* 64(2)**,** 104-117.

Sinek, J.P., Sanga, S., Zheng, X., Frieboes, H.B., Ferrari, M., and Cristini, V. (2009). Predicting drug pharmacokinetics and effect in vascularized tumors using computer simulation. *J Math Biol* 58(4-5)**,** 485-510.

Singer, S.J., and Nicolson, G.L. (1972). The fluid mosaic model of the structure of cell membranes. *Science* 175(4023)**,** 720-731.

Slamon, D., Clark, G., Wong, S., Levin, W., Ullrich, A., and McGuire, W. (1987). Human breast cancer: correlation of relapse and survival with amplification of the HER-2/neu oncogene. *Science* 235(4785)**,** 177-182. doi: 10.1126/science.3798106.

Smart, E.J., Graf, G.A., McNiven, M.A., Sessa, W.C., Engelman, J.A., Scherer, P.E., et al. (1999). Caveolins, liquid-ordered domains, and signal transduction. *Mol Cell Biol* 19(11)**,** 7289-7304.

Sotirellis, N., Johnson, T.M., Hibbs, M.L., Stanley, I.J., Stanley, E., Dunn, A.R., et al. (1995). Autophosphorylation Induces Autoactivation and a Decrease in the Src Homology 2 Domain Accessibility of the Lyn Protein Kinase. *Journal of Biological Chemistry* 270(50)**,** 29773-29780. doi: 10.1074/jbc.270.50.29773.

Steinkamp, M.P., Low-Nam, S.T., Yang, S., Lidke, K.A., Lidke, D.S., and Wilson, B.S. (2014). erbB3 Is an Active Tyrosine Kinase Capable of Homo- and Heterointeractions. *Molecular and Cellular Biology* 34(6)**,** 965-977. doi: 10.1128/MCB.01605-13.

Stiles, J.R., and Bartol, T.M. (2001). Monte Carlo methods for simulating realistic synaptic microphysiology using MCell. *Computational neuroscience: realistic modeling for experimentalists***,** 87-127.

Storch, B., Meixlsperger, S., and Jumaa, H. (2007). The Ig-alpha ITAM is required for efficient differentiation but not proliferation of pre-B cells. *Eur J Immunol* 37(1)**,** 252-260. doi: 10.1002/eji.200636667.

Stundzia, A.B., and Lumsden, C.J. (1996). Stochastic Simulation of Coupled Reaction–Diffusion Processes. *Journal of Computational Physics* 127(1)**,** 196-207. doi: http://dx.doi.org/10.1006/jcph.1996.0168.

Stylianopoulos, T., and Jain, R.K. (2013). Combining two strategies to improve perfusion and drug delivery in solid tumors. *Proceedings of the National Academy of Sciences*. doi: 10.1073/pnas.1318415110.

Sven, K., and Josipa, F. (2007). *Interstitial Hydrostatic Pressure: a Manual for Students.*

Tannock, I.F., Lee, C.M., Tunggal, J.K., Cowan, D.S.M., and Egorin, M.J. (2002). Limited Penetration of Anticancer Drugs through Tumor Tissue: A Potential Cause of Resistance of Solid Tumors to Chemotherapy. *Clinical Cancer Research* 8(3)**,** 878-884.

Theilmann, R.J., Borders, R., Trouard, T.P., Xia, G., Outwater, E., Ranger-Moore, J., et al. (2004). Changes in water mobility measured by diffusion MRI predict response of metastatic breast cancer to chemotherapy. *Neoplasia* 6(6)**,** 831-837.

Therasse, P., Arbuck, S.G., Eisenhauer, E.A., Wanders, J., Kaplan, R.S., Rubinstein, L., et al. (2000). New Guidelines to Evaluate the Response to Treatment in Solid Tumors. *Journal of the National Cancer Institute* 92(3)**,** 205-216. doi: 10.1093/jnci/92.3.205.

Thoeny, H.C., and Ross, B.D. (2010). Predicting and monitoring cancer treatment response with diffusion-weighted MRI. *J Magn Reson Imaging* 32(1)**,** 2-16.

Threadgill, D.W., Dlugosz, A.A., Hansen, L.A., Tennenbaum, T., Lichti, U., Yee, D., et al. (1995). Targeted disruption of mouse EGF receptor: effect of genetic background on mutant phenotype. *Science* 269(5221)**,** 230-234.

Tolar, P., Hanna, J., Krueger, P.D., and Pierce, S.K. (2009). The Constant Region of the Membrane Immunoglobulin Mediates B Cell-Receptor Clustering and Signaling in Response to Membrane Antigens. *Immunity* 30(1)**,** 44-55. doi: https://doi.org/10.1016/j.immuni.2008.11.007.

Tonegawa, S. (1983). Somatic generation of antibody diversity. *Nature* 302(5909)**,** 575-581.

Tonkin, K., Tritchler, D., and Tannock, I. (1985). Criteria of tumor response used in clinical trials of chemotherapy. *Journal of Clinical Oncology* 3(6)**,** 870-875. doi: 10.1200/jco.1985.3.6.870.

Trageser, D., Iacobucci, I., Nahar, R., Duy, C., von Levetzow, G., Klemm, L., et al. (2009). Pre–B cell receptor–mediated cell cycle arrest in Philadelphia chromosome–positive acute lymphoblastic leukemia requires IKAROS function. *The Journal of Experimental Medicine* 206(8)**,** 1739-1753. doi: 10.1084/jem.20090004.

Treanor, B., Depoil, D., Gonzalez-Granja, A., Barral, P., Weber, M., Dushek, O., et al. (2010a). The Membrane Skeleton Controls Diffusion Dynamics and Signaling through the B Cell Receptor. *Immunity* 32(2)**,** 187-199. doi: https://doi.org/10.1016/j.immuni.2009.12.005.

Treanor, B., Depoil, D., Gonzalez-Granja, A., Barral, P., Weber, M., Dushek, O., et al. (2010b). The membrane skeleton controls diffusion dynamics and signaling through the B cell receptor. *Immunity* 32(2)**,** 187-199. doi: 10.1016/j.immuni.2009.12.005.

Tredan, O., Galmarini, C.M., Patel, K., and Tannock, I.F. (2007). Drug resistance and the solid tumor microenvironment. *J Natl Cancer Inst* 99(19)**,** 1441-1454. doi: 10.1093/jnci/djm135.

Tsang, E., Giannetti, A.M., Shaw, D., Dinh, M., Tse, J.K.Y., Gandhi, S., et al. (2008). Molecular Mechanism of the Syk Activation Switch. *Journal of Biological Chemistry* 283(47)**,** 32650-32659. doi: 10.1074/jbc.M806340200.

Tsourkas, P.K., Somkanya, C.D., Yu-Yang, P., Liu, W., Pierce, S.K., and Raychaudhuri, S. (2012). Formation of BCR oligomers provides a mechanism for B cell affinity discrimination. *J Theor Biol* 307**,** 174-182. doi: 10.1016/j.jtbi.2012.05.008.

Tunggal, J.K., Cowan, D.S., Shaikh, H., and Tannock, I.F. (1999). Penetration of anticancer drugs through solid tissue: a factor that limits the effectiveness of chemotherapy for solid tumors. *Clin Cancer Res* 5(6)**,** 1583-1586.

Tzahar, E., Waterman, H., Chen, X., Levkowitz, G., Karunagaran, D., Lavi, S., et al. (1996). A hierarchical network of interreceptor interactions determines signal transduction by Neu differentiation factor/neuregulin and epidermal growth factor. *Mol Cell Biol* 16. doi: 10.1128/mcb.16.10.5276.

Ubelhart, R., Bach, M.P., Eschbach, C., Wossning, T., Reth, M., and Jumaa, H. (2010). N-linked glycosylation selectively regulates autonomous precursor BCR function. *Nat Immunol* 11(8)**,** 759-765. doi: http://www.nature.com/ni/journal/v11/n8/abs/ni.1903.html - supplementary-information.

Ullrich, A., and Schlessinger, J. (1990). Signal transduction by receptors with tyrosine kinase activity. *Cell* 61(2)**,** 203-212. doi: http://dx.doi.org/10.1016/0092-8674(90)90801-K.

van 't Veer, L.J., and Bernards, R. (2008). Enabling personalized cancer medicine through analysis of gene-expression patterns. *Nature* 452(7187)**,** 564-570.

van Zon, J.S., and Ten Wolde, P.R. (2005). Green's-function reaction dynamics: a particle-based approach for simulating biochemical networks in time and space. *The Journal of chemical physics* 123(23)**,** 234910.

Vandecaveye, V., de Keyzer, F., Vander Poorten, V., Deraedt, K., Alaerts, H., Landuyt, W., et al. (2006). Evaluation of the larynx for tumour recurrence by diffusion-weighted MRI after radiotherapy: initial experience in four cases. *Br J Radiol* 79(944)**,** 681-687.

Vaught, D.B., Stanford, J.C., Young, C., Hicks, D.J., Wheeler, F., Rinehart, C., et al. (2012). HER3 Is Required for HER2-Induced Preneoplastic Changes to the Breast Epithelium and Tumor Formation. *Cancer Research* 72(10)**,** 2672-2682. doi: 10.1158/0008-5472.can-11-3594.

von Boehmer, H., and Melchers, F. (2010). Checkpoints in lymphocyte development and autoimmune disease. *Nat Immunol* 11(1)**,** 14-20.

Vonakis, B.M., Chen, H., Haleem-Smith, H., and Metzger, H. (1997). The Unique Domain as the Site on Lyn Kinase for Its Constitutive Association with the High Affinity Receptor for IgE. *Journal of Biological Chemistry* 272(38)**,** 24072-24080. doi: 10.1074/jbc.272.38.24072.

Wallasch, C., Weiss, F.U., Niederfellner, G., Jallal, B., Issing, W., and Ullrich, A. (1995). Heregulin-dependent regulation of HER2/neu oncogenic signaling by heterodimerization with HER3. *The EMBO Journal* 14(17)**,** 4267-4275.

Wang, Z., Kerketta, R., Chuang, Y.L., Dogra, P., Butner, J.D., Brocato, T.A., et al. (2016). Theory and Experimental Validation of a Spatio-temporal Model of Chemotherapy Transport to Enhance Tumor Cell Kill. *PLoS Comput Biol* 12(6)**,** e1004969. doi: 10.1371/journal.pcbi.1004969.

Wei, S., Liu, L., Zhang, J., Bowers, J., Gowda, G.A.N., Seeger, H., et al. (2013). Metabolomics approach for predicting response to neoadjuvant chemotherapy for breast cancer. *Molecular Oncology* 7(3)**,** 297-307. doi: https://doi.org/10.1016/j.molonc.2012.10.003.

Wiedeman, M.P. (1963). Dimensions of blood vessels from distributing artery to collecting vein. *Circ Res* 12**,** 375-378.

Wilson, B.S., Oliver, J.M., and Lidke, D.S. (2011). Spatio-temporal Signaling in Mast Cells. *Advances in Experimental Medicine and Biology* 716**,** 91-106. doi: 10.1007/978-1-4419-9533-9_6.

Wilson, K.J., Gilmore, J.L., Foley, J., Lemmon, M.A., and Riese Ii, D.J. (2009). Functional selectivity of EGF family peptide growth factors: Implications for cancer. *Pharmacology & Therapeutics* 122(1)**,** 1-8. doi: https://doi.org/10.1016/j.pharmthera.2008.11.008.

Wofsy, C., Torigoe, C., Kent, U.M., Metzger, H., and Goldstein, B. (1997). Exploiting the difference between intrinsic and extrinsic kinases: implications for regulation of signaling by immunoreceptors. *J Immunol* 159(12)**,** 5984-5992.

Wofsy, C., Vonakis, B.M., Metzger, H., and Goldstein, B. (1999). One Lyn molecule is sufficient to initiate phosphorylation of aggregated high-affinity IgE receptors. *Proceedings of the National Academy of Sciences of the United States of America* 96(15)**,** 8615-8620.

Wolfram Research (2008). "Mathematica, Version 8.0, Mathematics and Algorithms". (http://www.wolfram.com/learningcenter/tutorialcollection/MathematicsAndAlgorithms/MathematicsAndAlgorithms.pdf).

Yamashita, T., Mao, S.Y., and Metzger, H. (1994). Aggregation of the high-affinity IgE receptor and enhanced activity of p53/56lyn protein-tyrosine kinase. *Proc Natl Acad Sci U S A* 91(23)**,** 11251-11255.

Yang, S., Raymond-Stintz, M.A., Ying, W., Zhang, J., Lidke, D.S., Steinberg, S.L., et al. (2007a). Mapping ErbB receptors on breast cancer cell membranes during signal transduction. *J Cell Sci* 120(Pt 16)**,** 2763-2773. doi: 10.1242/jcs.007658.

Yarden, Y. (2001). The EGFR family and its ligands in human cancer: signalling mechanisms and therapeutic opportunities. *European Journal of Cancer* 37, Supplement 4**,** 3-8. doi: https://doi.org/10.1016/S0959-8049(01)00230-1.

Yarden, Y., and Sliwkowski, M.X. (2001). Untangling the ErbB signalling network. *Nat Rev Mol Cell Biol* 2(2)**,** 127-137. doi: 10.1038/35052073.

Yuan, Y. (2016). Spatial Heterogeneity in the Tumor Microenvironment. *Cold Spring Harbor Perspectives in Medicine* 6(8). doi: 10.1101/cshperspect.a026583.

Zhang, J., Billingsley, M.L., Kincaid, R.L., and Siraganian, R.P. (2000). Phosphorylation of Syk Activation Loop Tyrosines Is Essential for Syk Function: AN IN VIVO STUDY USING A SPECIFIC ANTI-Syk ACTIVATION LOOP PHOSPHOTYROSINE ANTIBODY. *Journal of Biological Chemistry* 275(45)**,** 35442-35447. doi: 10.1074/jbc.M004549200.

Zhang, K., Wong, P., Duan, J., Jacobs, B., Borden, E.C., and Bedogni, B. (2013). An ERBB3/ERBB2 oncogenic unit plays a key role in NRG1 signaling and melanoma cell growth and survival. *Pigment Cell Melanoma Res* 26(3)**,** 408-414. doi: 10.1111/pcmr.12089.

Zhang, Y., Opresko, L., Shankaran, H., Chrisler, W.B., Wiley, H.S., and Resat, H. (2009). HER/ErbB receptor interactions and signaling patterns in human mammary epithelial cells. *BMC Cell Biology* 10(1)**,** 78. doi: 10.1186/1471-2121-10-78.