9-12-2014

Spatial Stochastic Modeling of the ErbB Receptor Family

Meghan Pryor

Meghan Marie Pryor
Candidate

Chemical and Nuclear Engineering
Department

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

Approved by the Dissertation Committee:

Jeremy S. Edwards, Chairperson

Bridget S. Wilson

Diane S. Lidke

Andrew P. Shreve

Spatial Stochastic Modeling of the ErbB Receptor Family

by

MEGHAN MARIE PRYOR

B.Ch.E., Chemical Engineering, University of Delaware, 2010
B.S., Quantitative Biology, University of Delaware, 2010

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Engineering

The University of New Mexico
Albuquerque, New Mexico

July, 2014

DEDICATION

I dedicate this work to my amazing family and wonderful husband.

First and foremost, to my parents for always encouraging me to follow my dreams and never pressuring me in any one direction; for supporting me when I sprung a fifth year of college and then graduate school on them; and for [thinking they were] secretly checking up on me by checking in with Colin to make sure I was surviving. To my Mom, for being my best friend, and laughing and crying with me through the past 4 years. To my Dad for our long phone conversations that covered every topic under the sun and for calling me and continuously asking simple questions until hitting the topic I truly needed to talk about. To my brother Ryan, who relentlessly picked on me growing up and taught me it was ok to be different and do my own thing; for answering my questions when I started [attempting] to read his stash of books on relativity; for having a stash of books on relativity in general; and for giving me the confidence to move 2000 miles away from home.

To my husband Colin, for supporting me when I took this unexpected detour to graduate school; for agreeing to take a giant leap of faith and move to New Mexico with me 4 years ago; and for always being an unwavering support system, especially over the past 4 years.

♥ ♣ ♦ ♠

"Why, sometimes I've believed as many as six impossible things before breakfast."

– *Red Queen, Alice's Adventures in Wonderland - Lewis Carroll*

♥ ♣ ♦ ♠

ACKNOWLEDGEMENTS

I, of course, must first acknowledge and thank my advisor Dr. Jeremy Edwards and co-mentor, Dr. Bridget Wilson. To Jeremy, I will forever be grateful that you and Dr. Janet Oliver agreed to take a chance on me, on the word of a few colleagues. Your guidance and knowledge have helped me to grow leaps and bounds over the past 4 years. To Bridget, I have learned so much about many different elements of being a research scientist from you over the past 4 years. You are a true role model for women in science.

I next would like to acknowledge and thank my committee: Dr. Jeremy Edwards, Dr. Bridget Wilson, Dr. Diane Lidke, and Dr. Andrew Shreve. Your insight into my research methods and findings over the last few years has helped me grow and become more confident in my work and as a research scientist.

To Dr. Babatunde Ogunnaike, Dr. Prasad Dhurjati, and Dr. Stanley Sandler, thank you for insisting I go to graduate school when it was the last thing on my mind, and for calling Jeremy to make graduate school happen. To Dr. Adam Halasz, for answering every single one of my math and algorithm questions and for keeping me sane when I thought my head would explode from all the stress. To Dr. Michael Wester and Dr. Stanly Steinberg for including me in your meetings when I first started this journey. To Michael, thank you for your invaluable help and input on math concepts and programming. I contribute a large amount of my growth in programming skills to your help.

Thank you to my experimental buddies, Dr. Shalini Low-Nam and Dr. Mara Steinkamp, without whom I would not know nearly as much experimental biology. To Shalini, your willingness to show me the ropes and sit down with me when I first started gave me the confidence I needed to keep going through graduate school. To Mara, you picking up where Shalini left off has helped me more than I think you will ever know. And to the rest of the OWL lab students, post-docs, and techs, whose input over the last 4 years has been invaluable.

Finally I would like to acknowledge and thank the Spatial Temporal Modeling Center for funding my first year as a graduate student, the INCBN IGERT program for funding two more years of graduate school, and the Center for Nonlinear Studies at Los Alamos National Laboratory for funding me through the end of my graduate studies.

SPATIAL STOCHASTIC MODELING OF THE ERBB RECEPTOR FAMILY

by

Meghan Marie Pryor

B.Ch.E., Chemical Engineering, University of Delaware, 2010

B.S., Quantitative Biology, University of Delaware, 2010

Ph.D., Engineering, University of New Mexico, 2014

ABSTRACT

ErbB transmembrane receptors are a family of 4 receptor tyrosine kinases that interact with one another through homo and heterodimer interactions. When these dimers form, the kinase domains on the receptor tails interact with one another, transphosphorylating one another, initiating a signal cascade. The signaling pathways these receptors participate in are responsible for many different cell functions including apoptosis, growth, and proliferation. The overexpression of these receptors has been linked to various forms of cancer, emphasizing the importance of understanding how these receptors interact with one another to trigger these cascades. Single Particle Tracking experiments have provided more precise and detailed measures of dimer lifetimes and diffusion. A major observation from the experiments is the anomalous diffusion of the receptors. One suggested contributor to this anomalous diffusion is confinement zones on the membrane. In this work, we develop, validate, and implement a spatial stochastic model to study these receptors and uncover how their kinetics and dynamics as well as the membrane landscape come together to impact erbB activation.

We start by focusing on erbB1. Single particle tracking experiments show that receptor pairs interact repeatedly over a period of time. One possible explanation for

these repeated interactions is to facilitate phosphorylation. An asymmetric phosphorylation model is proposed, where one receptor in the dimer pair is responsible for activating the other receptor, the receiver, which then in turn phosphorylates the original activator. The model shows that the confinement zones on the membrane play a critical role in causing repeated receptor interactions and reveals that receptors dynamically switch between different activation states over time. Our work continues by delving deeper into the membrane landscape. Single particle tracking data is analyzed to investigate the characteristics of the observed anomalous diffusion. The analysis gives an estimate for the size range of the confinement zones and shows that they are a series of domains, not corrals. Taking the single particle tracking analysis one step further, we develop a Domain Reconstruction Algorithm that reconstructs confinement zone shapes and sizes from single particle tracking trajectories. In the final study, we move on to erbB2 and erbB3 interactions. ErbB3, which is traditionally believed to be kinase dead, has recently been shown to have weak kinase activity. Through kinase assay experiments, we show in the presence of erbB2 and heregulin, erbB3 has measurable kinase activity. Using the reconstructed domains from erbB2 and erbB3 data to create a simulation space, and experimental data from the kinase assay and single particle tracking, we extend the erbB1 spatial stochastic model for this study. We show that erbB2 and erbB3 have significantly different interactions with the cellular membrane confinement zones, erbB3 is dependent on erbB2 activation, and erbB3 homodimer stability inhibits erbB3 activation. Extension of the model to investigate mutation behaviors in erbB3 receptors reveals insights into how a gain of function mutation in the erbB3 kinase domain impacts erbB2 and erbB3 interactions. Finally, discovery of a gain of function mutation in the

kinase domain of erbB3 is connected to an uptick in erbB3 kinase activity. As a path forward from this work, we suggest using the spatial stochastic model to investigate more possible mutations in erbB3 receptors to give better insight into which mutations would be promising to explore.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

Cellular signaling is mediated through a host of protein interactions. Proteins on the cellular membrane interact, become activated, and initiate signal cascades, consisting of intracellular proteins, in the cytoplasm. These interactions have long been studied for various cellular proteins. Popular protein systems include the ErbB family implicated in various forms of cancer (Citri & Yarden, 2006; Yarden & Sliwkowski, 2001), the IgE receptor known for its role in allergies, and Toll-like receptors (TLR) that trigger immune responses (Akira & Takeda, 2004). A range of experimental procedures is used to study these systems, including immunoprecipitation and western blotting. More recently, high-resolution experiments have been developed that allow for greater molecular resolution. These experiments include single particle tracking (SPT) (Low-Nam et al, 2011), fluorescence resonance energy transfer (FRET), and transmission electron microscopy (TEM) (Radhakrishnan et al, 2012), to name a few.

Experiments give important insight into how different proteins interact with one another. These insights can be extended beyond current experimental capabilities by deriving a relevant mathematical model. Building off of the biological model pieced together using experiments and incorporating experimental data for model validation, the mathematical model can be used to visualize and investigate how multiple processes come together. A quick search of PubMed reveals the recent uptick in interest in integrating mathematics and biology in the last decade. Figure 1.1 shows the fraction of papers published on specific search topics from 1980–2013. Figure 1.1 (left) shows the trends for publications with topics based around mathematical modeling and biology,

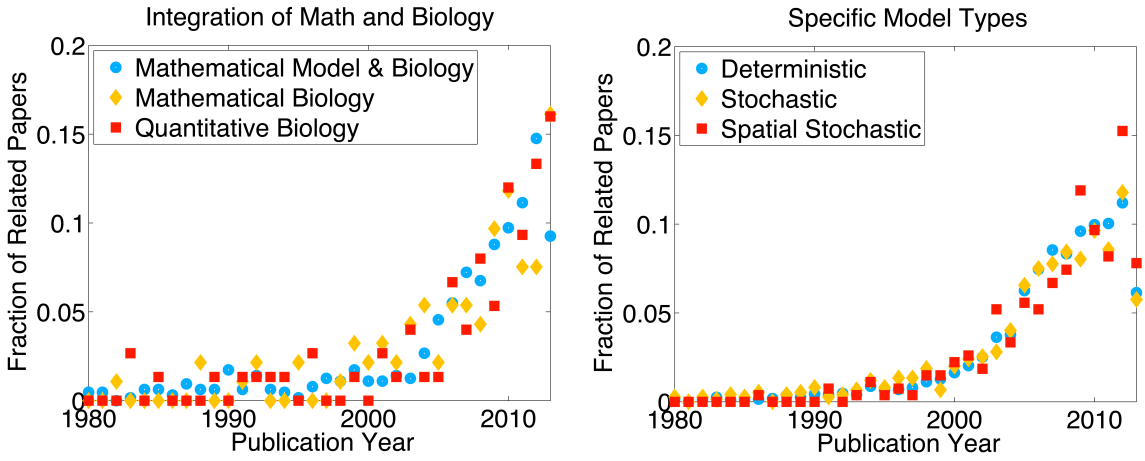

Figure 1.1: Papers published focusing on mathematics and biology. Fraction of papers published on specific search topics from 1980-2013.

mathematical biology, and quantitative biology. Figure 1.1 (right) shows the trends for publications with topics focused on different types of mathematical model methods also focused on biology. Based on the conclusions of the mathematical model, novel experimental methods can be developed to examine the mathematical model outcomes to further validate or modify the model. This methodology allows for an efficient cycle of experimentation, modeling, and novel experiment development.

1.2 MOTIVATION

In this work, we focus on the ErbB family of tyrosine receptor kinases (RTKs). ErbB receptors are a group of four tyrosine kinases (ErbB1/2/3/4) that are activated via ligand binding and subsequent formation of homo and heterodimers. The main function of the ErbB family is to mediate important cellular processes such as homeostasis, pathology, and development (Linggi & Carpenter, 2006) as well as the interactions between cells (Yarden & Sliwkowski, 2001). Specifically, ErbB receptors play an important part in the regulation of cell growth, proliferation, differentiation, and apoptosis (Baselga & Swain, 2009; Costa et al, 2009b). Given these important roles in normal cell development, it is not surprising that in the early 1980's ErbB1, ErbB2, and ErbB3 were implicated in the progression of cancers (Baselga & Swain, 2009; Yarden & Sliwkowski, 2001) including breast, lung, and bladder (Britten, 2004). ErbB4's role in cancer development is currently not well understood (Baselga & Swain, 2009). ErbB1, which can form dimers with all of the other receptors, typically forms mitogenic homodimers and is overexpressed in non-small-cell lung cancer (Yarden & Sliwkowski, 2001). ErbB3, which has classically been believed to be kinase dead (Berger et al, 2004; Linggi & Carpenter, 2006), dimerizes with ErbB2 to form an oncogenic heterodimer (Baselga & Swain, 2009). Both ErbB2 and

ErbB3 are prevalent in breast cancer (Baselga & Swain, 2009; Yarden & Sliwkowski, 2001). Due to the strong link between these receptors and various forms of cancers, it is crucial to gain a better understanding of how the receptors interact and their signaling mechanisms.

Traditionally, experimental methods are used to understand ErbB signaling. Immunoelectron microscopy (EM) has been used to study ErbB receptor distributions on the cellular membrane, as well as the adapter protein recruitment (Yang et al, 2007). More recently, Single particle tracking (SPT) experiments have been used to visualize ErbB behavior within the live cell context. This SPT data has provided improved dynamic measures of dimer lifetimes, receptor diffusion, and insight into different receptor pair combinations. Low-Nam et al. (2011) published ErbB1 SPT data showing strong evidence that transient receptor co-confinement promotes repeated interactions between ErbB1 monomers. Steinkamp et al. (2014) recently published new SPT that indicates ErbB3 does in fact form a homodimer, as well as other experiments showing that the kinase domain of ErbB3 is active. To investigate the observation made from the experiments on ErbB1 and ErbB2/3, the parameters from the SPT experiments must be used to develop a spatial stochastic model. This model will be able to resolve dynamics of receptors that cannot yet be investigated through experimental methods. In a step towards making the connection between experiments and modeling stronger, **the overall goal of this work is to develop and implement a spatial stochastic model using parameters from live cell imaging to investigate the kinetics and dynamics of ErbB receptors as influenced by the membrane landscape and other ErbB receptors.**

1.3 MATHEMATICAL SIMULATION OF MEMBRANE PROTEIN CLUSTERING FOR EFFICIENT SIGNAL TRANSDUCTION

1.3.1 Notes

The work shown in this section, **1.3**, was published in Annals of Biomedical Engineering in November 2012, Volume 40, Pages 2307-2318. DOI 10.1007/s10439-012-0599-z.

Krishnan Radhakrishnan[1], Ádám Halász[5], Meghan M. McCabe[3], Jeremy S. Edwards[2,3,4] and Bridget S. Wilson[1,4*]

[1]Dept. of Pathology, [2]Dept. of Molecular Genetics and Microbiology, [3]Dept. of Chemical Engineering and [4]Cancer Center, University of New Mexico, Albuquerque, NM [5]Dept. of Mathematics, West Virginia University, Morgantown, WV

*Address correspondence to bwilson@salud.unm.edu

1.3.2 Abstract

Initiation and propagation of cell signaling depends on productive interactions between signaling proteins at the plasma membrane. These diffusion-limited interactions can be influenced by features of the membrane that introduce barriers, such as cytoskeletal corrals, or microdomains that transiently confine both transmembrane receptors and membrane-tethered peripheral proteins. Membrane topographical features can lead to clustering of receptors and other membrane components, even under very dynamic conditions. This review considers the experimental and mathematical evidence that protein clustering impacts cell signaling in complex ways. Simulation approaches used to consider these stochastic processes are discussed.

1.3.3 Key Terms

Clustering; Spatial Stochastic Simulations; Cell Signaling

1.3.4 Introduction

Cell signaling, used for both intracellular and intercellular communication, is essential for the healthy physiological functioning of multi-cellular organisms. Ligand binding to a transmembrane receptor triggers an intracellular signaling cascade that results in altered cell behavior. The proper integration of different environmental signals is critically important to many biological processes, including cell survival, differentiation, proliferation and migration (Bublil & Yarden, 2007; Keating et al, 2008; Kitaura et al, 2003; Lo, 2010; Wennerberg et al, 2005; Yang et al, 2007). Aberrations in signal transduction have been implicated in numerous pathologies, from allergy and asthma to many different cancers (Bublil & Yarden, 2007; Colicelli, 2004; Govindan, 2010; Hynes & MacDonald, 2009; Keating et al, 2008; Lo, 2010; ten Klooster & Hordijk, 2007; Vigil et al, 2010; Yang et al, 2007). Signal transduction pathways have therefore been studied extensively, and many drugs developed to target them (Bublil & Yarden, 2007; Friday & Adjei, 2008; Govindan, 2010; Lo, 2010; ten Klooster & Hordijk, 2007; Vigil et al, 2010).

Knowledge of the structure of the plasma membrane and of signaling processes continues to improve, due to advances in experimental techniques and imaging technologies (Lidke & Wilson, 2009; Wells et al, 2010; Wilson et al, 2010). There is considerable evidence for the concept that the cell membrane is compartmentalized into microdomains, such as protein islands (Wilson et al, 2007) and lipid rafts (Nagy, 2002). Receptor clustering in small or large aggregates (illustrated schematically in Figure 1.2)

at discrete locations has been noted in many cell types (Abulrob et al, 2010; Bader et al, 2009; Hartman & Groves, 2011; Keating et al, 2008; Saffarian et al, 2007; Szabo et al, 2008; Yang et al, 2007), prompting intense interest in roles for membrane microdomains in signal propagation and preliminary mathematical studies to understand both formation of clusters and their role in cell signaling (Brinkerhoff et al, 2004; Costa et al, 2009a; Costa et al, 2009b; Fallahi-Sichani & Linderman, 2009; Hsieh et al, 2010; Hsieh et al, 2008; Linderman, 2009; Reddy et al, 2010; Tian et al, 2007; Tian et al, 2010). There is general agreement that the composition of these microdomains is quite heterogeneous and, further, that their stability is influenced by the dynamic interactions of the cortical cytoskeleton with membrane proteins and lipids. The cytoskeleton also limits diffusion of membrane constituents by forming "picket fences" and "corrals" (Kusumi et al, 2005b; Suzuki et al, 2005). The role of these membrane features in promoting or limiting protein-protein interactions remains controversial, since there is strong potential to both enhance and inhibit signaling (Allen et al, 2007; Costa et al, 2009a; Miura et al, 2001; Pike, 2003). To help resolve these issues, several groups are developing spatially realistic mathematical simulations of receptor motion, aggregation/clustering and activation in the cell membrane.

It is important to note that parameters for these mathematical models rely on powerful new experimental techniques. High resolution microscopy techniques, such as Transmission Electron Microscopy (TEM) and photoactivation Light Microcrosopy (PALM), have been applied to map the spatial distribution of signaling molecules in fixed cells (Lillemeier et al, 2010; Wilson et al, 2007). These snapshot images of protein distributions can be supplemented with powerful new live cell imaging approaches,

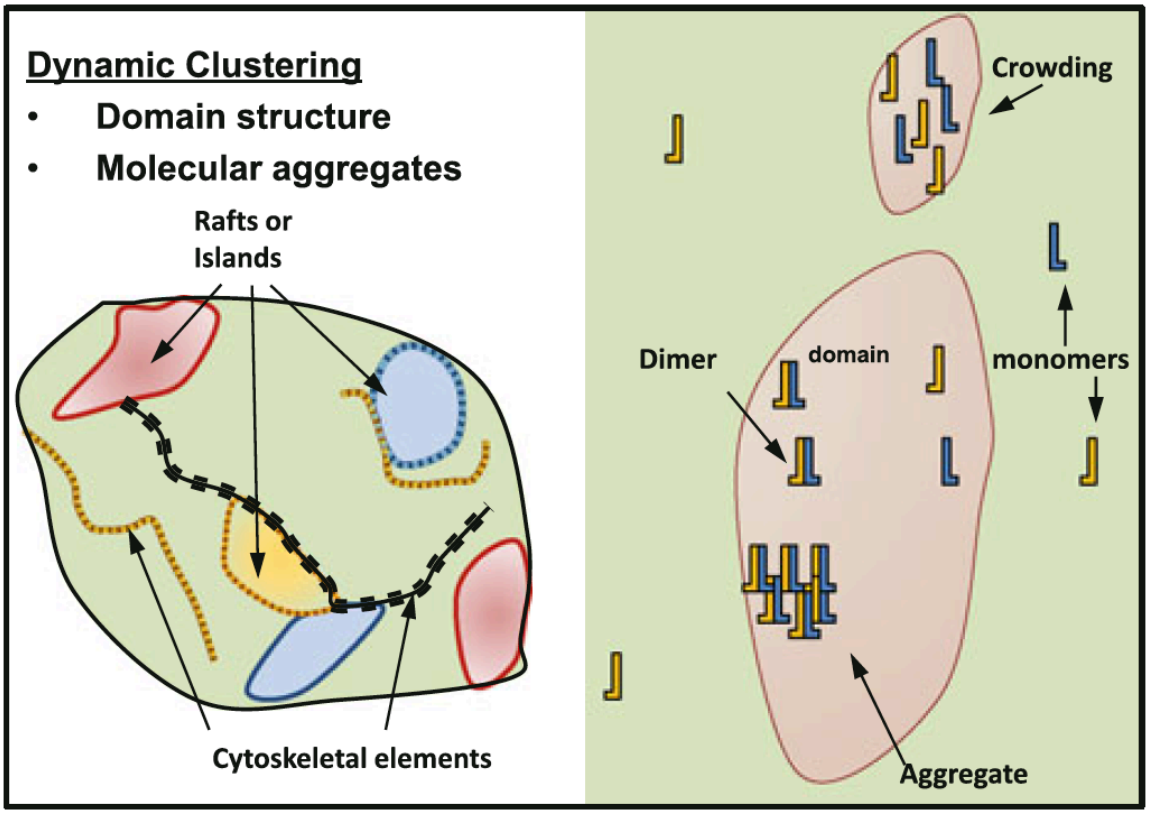

Figure 1.2: Schematic representation of microdomains and receptor clustering. Left: Cartoon representation of features that can subcompartmentalize the plasma membrane, including rafts or islands, and the cortical cytoskeletal network. These features are highly dynamic, permitting rapid exchange by diffusion. Right: Representation of the distribution of receptors (yellow, blue symbols) in and out of domains (pink regions) formed by these features. Arrows point to various states, including monomers, dimers, and aggregates. Receptors that are transiently trapped in domains are locally crowded (arrow, top right) and appear as clusters by imaging technologies. This molecular crowding can be more pronounced upon ligand stimulation, due in part to formation of dimers and larger aggregates with decreased diffusive mobility. This review considers the experimental and computational evidence that molecular crowding influences receptor dimerization/aggregation and recruitment of signaling proteins.

including fluorescence resonance energy transfer (FRET), fluorescence lifetime correlation spectroscopy (FLCS), and single particle tracking (SPT) experiments (Lidke & Wilson, 2009). These techniques can generate key information regarding the kinetics of protein-protein interactions, including rates of dimerization, size of receptor aggregates and changes in diffusion properties (Low-Nam et al, 2011). These rich data sets support the development of more accurate and detailed mathematical models that in turn improve understanding of biological results.

1.3.4.1 Key concepts and definitions relevant to the consideration of protein clustering in the plasma membrane

In this brief review, we focus attention on the mathematical simulation of protein clustering in the plasma membrane, an initial step in many signaling pathways. The protein species considered may be a surface receptor, that is triggered by binding to an extracellular ligand, or could be an intracellular signaling partner, such as an adaptor protein or enzyme that propagates signaling through the cell interior. We define clustering as the non-random spatial distribution of a membrane species, which can be observed and experimentally validated through modern technologies. “Snap-shot” images of membrane proteins often capture some level of clustering even before the onset of ligand binding to receptors or active signaling (Bader et al, 2009). We hypothesize that these basal levels of clustering arise from brief, non-productive interactions among proteins as they encounter one another while diffusing in the plasma membrane or when proteins are transiently co-confined in a raft, island, or corral (Figure 1.2). Thus clustering in this sense is not synonymous with oligomerization, which reflects the direct and measurable interaction between membrane components. It is important to point out that stable oligomers cannot be distinguished from unstable clusters in imaging

techniques using fixed cells, such as TEM and PALM. However, new imaging protocols can now accurately measure the dynamics of protein-protein interactions at the molecular scale (Lidke & Wilson, 2009). A recent example from our Center is the simultaneous single particle tracking (SPT) of pairs of EGFR molecules, each labeled by virtue of binding to EGF conjugated to different colors of quantum dot probes; only when two EGF-QD-bound receptors were both coincident and exhibited correlated motion, could they pass the stringent criteria for oligomerization (Low-Nam et al, 2011). The concept of clustering becomes particularly important as we consider the data suggesting that the spatial proximity of proteins can *promote* protein-protein interactions, including oligomerization, by increasing the likelihood of productive collisions.

1.3.4.2 Choosing the right modeling approach

Mathematical models constructed to date to study signal transduction pathways are of varied complexity. They can be classified conveniently as deterministic methods, in which inherent temporal and spatial fluctuations in diffusion and reaction rates are ignored, and stochastic methods, which attempt to capture these fluctuations (Figure 1.3). The simplest modeling approach is to assume that the system of interest is well mixed, without any spatial concentration gradients, and describe the reactions by a system of ordinary differential equations (ODEs). The utility of ODE modeling is enhanced by systematic sensitivity analysis, which examines automatically changes in model behavior due to parameter variation (Radhakrishnan, 1991; Radhakrishnan et al, 2009). Such a deterministic, well-mixed approach continues to be widely used (Tian et al, 2010), and has produced useful results (Brightman & Fell, 2000; Radhakrishnan et al, 2009). However, these approaches do not take into account either spatial inhomogeneities or

stochastic fluctuations, which can be significant when the number of molecules in the region of interest is small. At a slightly higher level of complexity, some spatial description is provided by dividing the region of interest into separate well-mixed compartments. Additional ODEs are needed to describe inter-compartmental species translocation reactions, thus mimicking spatial movement.

These well-mixed, ODE-based continuum pathway models (Kholodenko et al, 1999) were expanded to include spatial inhomogeneity (Brown & Kholodenko, 1999; Slepchenko et al, 2003) by solving partial differential equations (PDEs) that include molecular diffusion effects. Stochastic methods that assume spatially well mixed systems have also been developed to account for temporal fluctuations (Gillespie, 2007; Li et al, 2008). Stochastic PDEs include both spatial information and temporal fluctuations. The most detailed, and thus most complex, models are fully spatial, stochastic methods that track the movement of individual molecules (Andrews & Bray, 2004; Burrage et al, 2007; Costa et al, 2009a; Costa et al, 2009b; Grima & Schnell, 2008; Hsieh et al, 2010; Hsieh et al, 2008; Reddy et al, 2010; Tolle & Le Novere, 2010b). However, the computational burden increases rapidly with increasing complexity of the modeling approach. Figure 1.3 summarizes the various modeling approaches and their range of applicability.

Mathematical simulation of events in the plasma membrane faces unique challenges. Membrane proteins are constantly undergoing random motion in the plane of the membrane, where the diffusion rate is influenced by the environment, such as hindrance by microdomains, and thus varies both spatially and temporally. Optimally, the spatial location of every protein needs to be predicted, in order to capture clustering imposed by

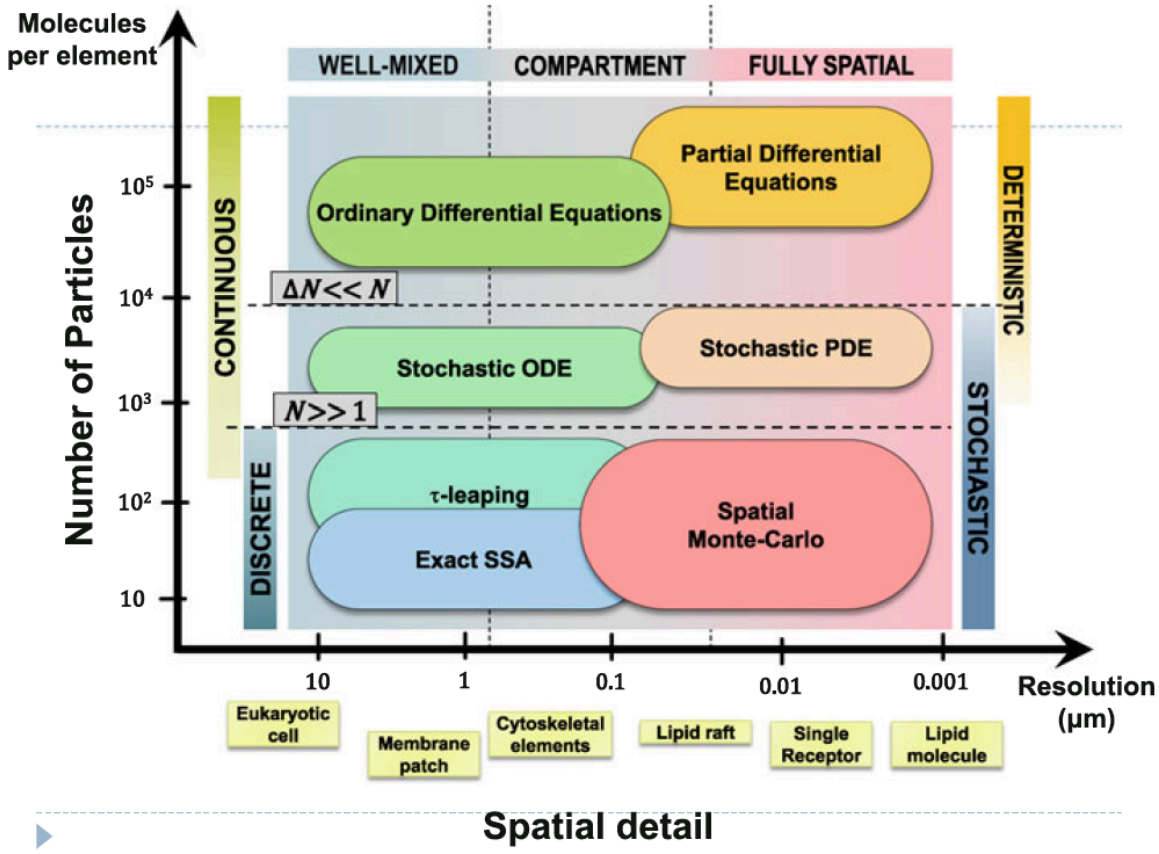

Figure 1.3: Classes of mathematical models for molecular processes in cells and the scales at which they are applicable to signaling processes. A possible quantitative guide is the size of the largest element that can be treated as spatially homogeneous (horizontal axis) and the typical number of molecules of one species in the element (vertical axis). The suggested spatial resolution is determined by the size of the biological element of interest and current computational capabilities. Spatially resolved models are resource-intensive, and are therefore generally applied to small subsystems. Cell-level models of large signaling networks are typically well mixed; spatial Monte Carlo studies rarely scale beyond a few hundred nanometers. A promising approach for multiscale applications is a combination of compartment-based models at the large scales and fully spatial simulations focused on a few important processes within small structural elements of the membrane. Temporal fluctuations arise largely from the discrete and stochastic nature of the underlying molecular processes. The relative magnitude of temporal fluctuations (DN) decreases as the number of particles increases. The discrete nature of the particle number can thus be ignored when N is significantly greater than 1. That is, deviations from the expected average behavior can be neglected when the expected magnitude of the fluctuations is small compared to N.

membrane topography and to predict the outcomes of both transient and prolonged protein-protein binding events. Fully spatial, stochastic methods offer capabilities that can capture accurately the dynamics of these events, but can be associated with prohibitively high computation cost. Novel hybrid approaches show promise for solving some of these computational challenges.

Finally, this section would not be complete without introducing the unique power of rules-based approaches (Faeder et al, 2009; Hlavacek et al, 2006). Here, molecular interactions in signaling networks are treated as systems of encoded rules. Molecules are represented as structural objects that have modular domains and associated states representing conformations or covalent modifications of these domains. Importantly the input files and model specification blocks are compatible with multiple types of computational approaches, including coupled ODEs that result in deterministic solutions of reaction kinetics as well as stochastic methods.

1.3.5 Applications in specific signaling pathways

Sections below briefly summarize mathematical models that have been developed to study signal transduction pathways, with emphasis on methods developed by our group and others to capture the influence of clustering and other spatial aspects. We focus on three representative signal transduction pathways (EGFR, Ras/MAPK and GPCR) where protein clustering has been implicated, and on the modeling approaches used to approach this unique set of challenges.

1.3.5.1 Our group's focus: spatial aspects of signaling through the epidermal growth factor receptor

A member of the ErbB family of plasma membrane receptors, EGFR is critically important to many biological processes, including embryonic development and carcinogenesis (Bublil & Yarden, 2007; Keating et al, 2008; Yang et al, 2007). Upon binding any one of several ligands, including EGF, the ErbB receptors homo- or hetero-dimerize. Dimerization is followed by transphosphorylation of tyrosine residues in receptor cytoplasmic tails, which enables recruitment of cytosolic signaling proteins. The reader is referred to Figures 2,3 in the article by Telasco & Radhakrishnan, for diagrams of EGFR/ErbB1 dimerization, phosphorylation and adaptor protein recruitment (Telesco & Radhakrishnan, 2012). Subsequently, these complexes activate many different signaling cascades, including the Ras-MAPK pathway discussed in the next section.

There exists considerable experimental evidence for preexisting clusters of resting EGFR (Figure 1.4) and for dynamical changes after addition of ligand (Abulrob et al, 2010; Bader et al, 2009; Keating et al, 2008; Saffarian et al, 2007; Szabo et al, 2008; Yang et al, 2007). We have built simulation platforms at different levels of complexity, in order to evaluate the impact of EGFR clustering in the plasma membrane.

1.3.5.1.1 Approaches and methodology

Our first attempt to develop a spatial model of the EGFR pathway was a simple compartmental model that accounted for receptor density differences observed in the plasma membrane, with some regions having high-receptor density and others displaying low-receptor density (Mayawala et al, 2005b). The focus of this study was to explore whether the added computational complexity associated with spatial modeling was justified. Our initial goal was to determine if the non-uniform receptor distribution in the

cell membrane could account for the experimentally observed, concave-up Scatchard plot for binding of EGF ligand to its receptor. We simply optimized the distribution of receptors into high- and low-density regions, and were able to determine the parameter space that allowed for a concave-up Scatchard plot. This first attempt at compartmentalized spatial modeling showed that accounting for the spatial organization of receptors was highly valuable, and should be pursued, to enable both qualitative and quantitative understanding of cell signaling involving (at least) the EGFR.

This study convinced us of the utility of spatial modeling of membrane-bound receptors and of its importance in understanding cell signaling. We have now accumulated extensive experience in developing spatially realistic simulations of the cell membrane and also addressed the initiation of signaling (Chatterjee et al, 2005; Costa et al, 2009a; Costa et al, 2009b; Hsieh et al, 2010; Hsieh et al, 2008; Mayawala et al, 2004; Mayawala et al, 2005a; Mayawala et al, 2005b; Mayawala et al, 2006). Next, we summarize our development of lattice-based and lattice-free (or off-lattice) methods, as well as our use of hybrid approaches.

1.3.5.1.2 Lattice-based Monte Carlo (MC) approaches.

In lattice-based models, molecules are located at discrete grid points in the spatial domain and diffusion is restricted to movement to an unoccupied neighboring point. Lattice-based MC simulations have become very popular in the physics, chemistry, materials and engineering communities, as they provide spatio-temporal information at significantly reduced computational cost, compared to off-lattice simulations

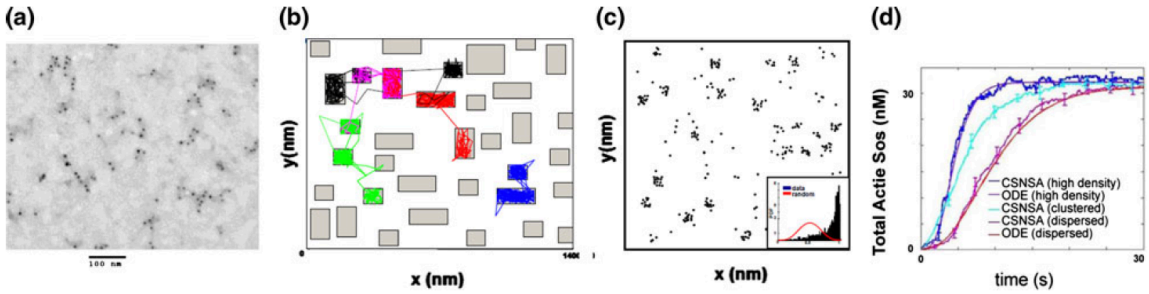

Figure 1.4: Experimental results and mathematical model predictions of EGFR clustering. (a) Experimental evidence for EGFR clustering in absence of ligand. Electron micrograph of gold particle-labeled EGF receptors in resting A341 cells (~2 million EGFR/cell), reveals a non-random distribution and provides evidence for receptor co-confinement. (b) Spatial domain used in lattice-free Monte Carlo simulation. The spatial domain simulated by the off-lattice Monte Carlo procedure was a square of area 2 lm$_2$, representative of a small region in the plasma membrane. This region was modified to include many islands or preferred domains (the gray rectangles within the membrane patch), to simulate the receptor-trapping microdomains seen in (a). Movement of receptors into and out of the simulated microdomains over a time period of 30 s is indicated by the thin colored tracings. Receptor trapping in the microdomains was reproduced mathematically by stipulating that receptors had a greater probability of entering these regions than of leaving them. (c) Simulation predictions of receptor clustering in absence of ligand. The predicted particle positions after 30 s of simulation time are indicated by the black dots. The Hopkins statistical test (inset) was used to test the randomness of receptor distribution. The right shift of the distribution (compared to the random or normal distribution shown in red) confirms the clustered nature of the receptors. The predicted receptor distribution compares well with the experimental observation in (a). (d) Simulations using a coupled spatial/nonspatial stochastic algorithm (CSNSA) support the conclusion that EGFR clustering promotes activation of the adaptor SOS. ODE models confirm this conclusion, using a fast diffusion coefficient to override contributions from membrane spatial organization (from Hsieh et al. and Costa et al.).

(Auerbach, 2000; Chuan Kang & Weinberg, 1995; Coppens et al, 1999; Gilmer, 1980; Zhdanov & Kasemo, 1994). The MC method is a coarse graining of molecular dynamics (MD) simulations (Auerbach, 2000), because MD is impractical for rare event dynamics, such as hopping between deep minima of a potential energy surface. The MC method stochastically solves an underlying master equation using pseudo-random numbers by constructing the probability with which the various states of the system have to be weighted according to a Markov process. MC simulations can provide continuous time information. Gillespie established the foundations of time-dependency for chemical reactions in a spatially homogeneous system (Gillespie, 1977; Gillespie, 2007). His approach is easily applicable to arbitrary complex computational systems, and is often referred to as the *kinetic or dynamic MC* method. Despite important algorithmic implementations (e.g., dependency graphs (Gibson & Bruck, 2000), lists of neighbors, binary-tree search, etc.), MC simulations are seriously plagued by (1) the presence of fast reactions that occur in large biochemical networks seen in biology and (2) the execution of one event at a time.

Our Spatial Kinetic Monte Carlo (SKMC) method (Mayawala et al, 2005a; Mayawala et al, 2005b) utilizes a modified null-event, lattice-based MC algorithm (Costa et al, 2009a; Mayawala et al, 2006). The spatial domain, representing a small region of the plasma membrane, is a two-dimensional square lattice of side ℓ, divided into a large number of much smaller square bins of side a ($<< \ell$). The SKMC algorithm consists of first randomly selecting an occupied lattice site, and then choosing either a successful (reaction or diffusion) or unsuccessful (null) event, based on calculated probabilities. If a

successful event is chosen, it is executed. The transition rate $\Gamma^{d}_{i\to j}$ for diffusion of species from any site *i* (i.e., lattice point *i*) to a nearest-neighboring site *j* as defined as

$$\Gamma^{d}_{i\to j} = \frac{1}{4}\Gamma^{D}\sigma_i\left(1-\sigma_j\right),\ j\in B_i$$

where $\Gamma^{D} = 4D/a^2$ and *D* is the diffusion coefficient of the species located at site *i*. The term B_i denotes the set of four possible nearest-neighboring sites to which diffusion can occur in two dimensions from site *i*. Because species are allowed to diffuse only to an unoccupied site, we define an occupancy function σ_j for each of the four nearest-neighboring sites, in order to simplify the procedure for computing the transition rate for diffusion. For any site k (= *i* or *j*), σ_k is set equal to 1 if the site is occupied, or to 0 if the site is unoccupied. The transition rate for a chemical reaction at site *i*, Γ^{r}_{i} depends on the reaction type and is directly related to the standard reaction rate.

The probability p^{x}_{i} of an event *x* (=*r* reaction or *d* diffusion) at site *i* is computed by using the relation

$$p^{x}_{i} = \frac{\Gamma^{x}_{i}}{\Gamma_{\max}}$$

where $\Gamma_{\max}$ is a normalization constant, defined as

$$\Gamma_{\max} = 4\left(\frac{\Gamma^{D}}{4} + \max\left(\sum_{allforwardreactionevents}\Gamma^{r}\right)\right)$$
$$+\max\left(\sum_{allbackwardreactionevents}\Gamma^{r}\right)'$$

where the multiplicative factor of 4 accounts for events occurring in the four directions of the two-dimensional square lattice. Finally, the time step Δt used to advance the simulation time is computed as $\Delta t = 1/\Gamma_{\max}$.

1.3.5.1.3 Rule-based, non-lattice simulator

Our non-lattice, stochastic simulator is an alternative approach (Hsieh et al, 2010; Hsieh et al, 2008). In the lattice-free method, particles are not confined to discrete points in space but are randomly repositioned upon undergoing a diffusion event. Receptors and other proteins in the 2D membrane and 3D cytosolic space are represented by sphere-like particles with radii determined from experimental data and their coarse-grained molecular models. At each time step, species diffusion is simulated as Brownian motion (Figure 1.4). In addition, species have the potential to react with spatially nearby species. This simulator was designed for flexible model development and deployment by a modularized and rule-based approach. It tracks the individual reactions of multistate molecules and accommodates complex situations.

1.3.5.1.4 Hybrid approaches

We continue to improve our basic SKMC algorithm, leading to increased efficiency and speed of the simulations. One significant advance was the coupling of our lattice-based SKMC simulations on the cell membrane to well-mixed stochastic simulations within the cytosol (Costa et al, 2009a). In Costa et al (2009a) we describe the development of an algorithm that couples a spatial stochastic model of membrane receptors with a nonspatial stochastic model of cytosolic reactions. Our novel hybrid algorithm provided a computationally efficient method to evaluate the effects of spatial heterogeneity on the coupling of receptors to cytosolic signaling partners. Results obtained using a compartmental ODE method compared well with those generated with our hybrid model. Thus, for sufficiently high receptor copy number, the far simpler ODE model may be adequate. However, for spatially inhomogenous systems where the

receptors numbers are low, the hybrid method was significantly superior to the ODE model.

1.3.5.1.5 EGFR density, through clustering or overexpression, influences signaling output

We have applied these methods to study the early molecular mechanisms involved in EGFR signaling. For example, we applied the lattice-based spatial stochastic model of the plasma membrane to examine the influence of cytoskeletal corral openings on EGFR clustering (Costa et al, 2009b). Clustering was shown to depend on both receptor concentration and picket fence density. For high picket fence densities, clustering increased with increasing receptor concentration in the range examined. Conversely, low receptor concentrations combined with small corral sizes inhibited clustering; at normal to high receptor concentration, maximal clustering occurred at an intermediate corral size (~100 nm). These results indicate that both the number of clusters and the average cluster size are likely to be complex functions of receptor density and microdomain size. It follows that compartmentalization of the plasma membrane could either *inhibit* or *enhance* signaling, concepts that require further exploration.

The non-lattice, rules-based simulator allowed us to explore the effect of EGFR overexpression and its relation to carcinogenesis (Hsieh et al, 2008). We postulated that increased receptor density in membrane microdomains or protein islands might lead to more frequent interactions between non-ligand bound receptors, and further, that large numbers of these short-lived interactions might explain EGFR signaling known to occur even in the absence of ligand (Bader et al, 2009). One important aspect was consideration of EGFR extracellular domain conformation, based upon structural studies showing that

the resting EGFR is predominantly in a "closed" conformation. Binding of ligand is proposed to stabilize the extended conformation and expose the dimerization arm. In our simulations, we assumed that the resting EGFR "fluxes" between the open and closed states, but spends 99% of its time in the closed state. This property translates to a low probability that two diffusing monomers will collide under conditions where both expose their dimerization arms and are therefore competent to form a complex. The 2D simulation space included membrane microdomains that transiently trapped receptors (as in Figure 1.4), setting up clusters undergoing dynamic exchange. Remarkably, at levels of receptors typical of most normal cells, co-confinement in membrane microdomains lowered the threshold for ligand-independent receptor dimerization but resulted in very modest signaling output. When the simulation space was populated with densities typically seen in tumors with EGFR gene amplification, which can express millions of EGFR per cell, the percent of activated receptors could exceed 10% with our parameter values. Clustering had little effect in these cases, since the overall density on the membrane was already very high.

We have used both lattice and non-lattice models to consider how spatial aspects might affect the recruitment of signaling molecules to the phosphorylated EGFR tail (Costa et al, 2009b; Hsieh et al, 2010). In Hsieh et al (2010), we also considered the combinatorial complexities associated with the fact that EGFR has multiple phosphorylation sites, and further, the fact that each phosphotyrosine site is capable of binding multiple partners. We used coarse-grained molecular docking simulations to show that steric hinderance can impose important constraints on the composition of adaptor proteins capable of docking simultaneously on the EGFR tail. Modeling

predictions in Hsieh et al (2010) were quantitatively consistent with experimental data for the kinetics of both EGFR phosphorylation and recruitment of adaptor proteins. *Importantly, both papers provide mathematical support for the conclusion that clustering of receptors can amplify signaling by promoting sequential binding of adaptor proteins.* These results provide confidence in our models, and have led to ongoing studies of other growth factor receptors that initiate signaling through dimerization, particularly VEGFR, as well the heterodimerizing members of the ErbB family. This field continues to advance, as demonstrated by the hybrid approaches of Radhakishnan and colleagues (Telesco & Radhakrishnan, 2012) that consider ErbB structural and diffusion properties using increasingly complex models. Additional aspects of cell surface topography, such as the induction of membrane curvature by endocytic adaptor proteins, are new concepts that will provide important insight into the control of signal transduction through the biophysical principals of membranes.

1.3.5.2 Work by others: the case of signaling via Ras/MAPK pathways.

The Ras superfamily consists of over 100 small GTP-binding proteins (or GTPases), which respond to various extracellular stimuli to regulate important signal transduction pathways (Vigil et al, 2010; Wennerberg et al, 2005). These proteins, which have low intrinsic GTPase activity, "switch" between active GTP-bound and inactive GDP-bound conformations. The processes mediated by GTPases include cell division, differentiation, apoptosis and migration, cytoskeletal reorganization, and intracellular protein trafficking (ten Klooster & Hordijk, 2007). Abnormalities in these pathways are seen in various pathologies, including obesity, diabetes, inflammatory diseases, cardiovascular disease, neurological disease, and cancer (Colicelli, 2004; ten Klooster & Hordijk, 2007; Vigil et

al, 2010). Therefore the pharmacological targeting of GTPases and/or their signaling pathways is an active field (Vigil et al, 2010).

The Ras/Raf/MEK/ERK mitogen-activated protein kinase (MAPK) pathway has been investigated extensively, both in the clinic and the laboratory, and by mathematical modeling (Brightman & Fell, 2000; Friday & Adjei, 2008; Fujioka et al, 2006; Hatakeyama et al, 2003; Hornberg et al, 2005; Kholodenko et al, 1999; Kholodenko et al, 2010; Orton et al, 2005; Radhakrishnan et al, 2009; Sasagawa et al, 2005; Schoeberl et al, 2002; Tian et al, 2007; Tian et al, 2010; Wiley et al, 2003). Activation of a number of receptors, including EGFR, leads to guanine nucleotide exchange (dissociation of GDP, gain of GTP) by membrane-tethered Ras, thereby activating it. The activated Ras in turn activates Raf (Ras-associated factor), the first kinase in the cascade. Subsequently, Raf activates MEK (MAPK/extracellular signal-regulated kinase kinase), which then activates ERK (extracellular signal-regulated kinase). The translocation of phosphorylated ERK to the nucleus and activation of transcription factors mediates many cellular activities.

Numerous mathematical models have been developed to study this pathway (Brightman & Fell, 2000; Fujioka et al, 2006; Hatakeyama et al, 2003; Hornberg et al, 2005; Kholodenko et al, 1999; Kholodenko et al, 2010; Orton et al, 2005; Radhakrishnan et al, 2009; Sasagawa et al, 2005; Schoeberl et al, 2002; Tian et al, 2007; Tian et al, 2010; Wiley et al, 2003). Much of this work uses compartmental models and ODEs to follow the temporal evolution of activated ERK, and does not consider clustering in the plasma membrane. However, Tian et al (2007; 2010) have mathematically evaluated various spatial aspects of Ras signaling, including clustering in the plasma membrane. This group

utilized a hybrid approach to simulate reactions in the cell membrane and those in the cytosol, enabling them to separate the contribution of the plasma membrane structure to the signal. They combined the well-mixed stochastic model of Gillespie (Gillespie, 1977; Gillespie, 2007) to simulate reactions in the membrane with an ODE model for the cytosolic reactions. They assumed that the number of RasGTP clusters was proportional to the EGF concentration, and these clusters served as platforms for recruiting Raf to the plasma membrane for activation. The lifetime of RasGTP clusters was assumed to be normally distributed over a measured value. Plasma membrane reactions, in addition to binding and activation of Raf by RasGTP clusters, included recruitment by activated Raf of the KSR-MEK-ERK complex from the cytosol and activation of MEK by activated Raf and of ERK (MAPK) by activated MEK. KSR (kinase suppressor of Ras) is a scaffold protein that facilitates MAPK activation by providing binding sites for assembly of the signaling complex. The recruitment of both Raf and the KSR-MEK-ERK complex was modeled as occurring through random collisions with the plasma membrane. With dissolution of a nanocluster, all recruited proteins diffused back to the cytosol, where the activated MEK and ERK continued their roles. Using this model in conjunction with biological experiments, Tian et al. (2007) concluded that RasGTP clustering is essential for signal transduction. Moreover, the RasGTP clusters operate as sensitive switches in that they produce approximately the same levels of normalized activated ERK over a wide range of ligand concentration. One possible explanation for this behavior is the establishment of locally high concentrations of recruited proteins and thus the spatial restriction of active ERK production to RasGTP nanoclusters, whose generation and lifetime are themselves strictly regulated (Tian et al, 2007). Tian et al. (2007) also

concluded that the production of RasGTP nanoclusters in direct proportion to ligand concentration can ensure high fidelity of signal transduction.

Subsequently, Tian et al. (2010) incorporated models for following the temporal evolution of RasGTP clusters in the cell membrane. In particular, they studied K-Ras clustering and how it is influenced by the protein Galectin-3 (Gal3). Previous experimental work had shown that Gal3 is a scaffolding protein recruited to the plasma membrane, where it is necessary for the formation of Ras nanoclusters (Shalom-Feuerstein et al, 2008). Their mathematical model considered the two species, membrane-bound RasGTP and Gal3, initially in the cytosol. Once Gal3 is recruited by RasGTP, the RasGTP-Gal3 complexes are assumed to diffuse randomly in the plasma membrane and react with one another to form complexes of various sizes.

To simplify the calculation procedure, Tian et al. (2010) allowed for a maximum cluster size of ten. The various combinations of possible complexes resulted in a total of 27 species and 136 reactions in the plasma membrane. In agreement with our earlier observation, they concluded that spatial stochastic modeling of such a large system poses a considerable computational burden. Therefore they developed an ODE system to follow the temporal evolution of complexes of size 1-10, using a spatial stochastic model to only deduce collision rates among the complexes (Hsieh et al, 2010; Hsieh et al, 2008). This deterministic system was solved with a Runge-Kutta method suitable for stiff ODEs (Radhakrishnan, 1991). The collision rates were obtained by initially placing RasGTP randomly in a square-shaped representation of the plasma membrane. Recruitment of Gal3 produces the RasGal complex. These molecules were allowed to diffuse randomly, and a collision was said to occur when the distance between two molecules was less than

the sum of their radii. The collisions produced various combinations of Ras-Gal complexes. When a nanocluster, defined as a cluster consisting of 5 or more RasGTP molecules, formed it was assumed to become immobile in the plasma membrane. During the calculation procedure the total numbers of collisions giving rise to all cluster types were tracked. At the end of the computational time period, the collision rate constants were computed from the total numbers of collisions. Kinetic rate constants for the ODE model were then derived from the collision rate constants, by using a genetic algorithm in conjunction with experimental data. The validity of this deterministic ODE model was checked with results generated with a stochastic simulation algorithm (Gillespie, 1977). Presumably due to the large numbers of proteins, the stochastic simulations predicted only small fluctuations. This observation supports use of deterministic models when the protein copy number is high, in agreement with our observations.

Using this modeling approach, Tian et al. (2010) studied clustering of K-Ras-GTP in the plasma membrane arising from interactions with Gal3 for various KRas and Gal3 copy numbers. The simulation time period was sufficiently long for the system to equilibrate. The time to equilibrate was approximately two minutes, an important result because it is in good agreement with the time period required for RasGTP loading in response to stimulation (Tian et al, 2007). Their results also successfully reproduced the experimental results of Plowman et al. (2005) that approximately 42% of the RasGTP were in clusters and the average cluster size was approximately 7. Tian et al. (2010) also generated the equilibrium nanocluster number versus size histogram. Their results showed that nanoclusters with two to four molecules accounted for only 2.1% of the RasGTP, whereas a cluster size of 5 was the most prevalent. Nanoclusters larger than 5 in

size were progressively smaller in number, approximately inversely proportional to the size. The authors speculate that one possible reason for the lowered incorporation of RasGTPGal3 complexes into clusters of size 5 or larger is the remodeling of the lipid environment of the cluster by the stable pentamer. Their results also suggest that cluster formation is only weakly dependent on RasGTP concentration, and is determined by the Gal3 cytosolic concentration. Tian et al. (2010) concluded that on the basis of their simulations neglecting the formation of clusters with more than 10 RasGTP molecules is reasonable. Notably, this work illustrates the difficulty of spatial modeling of systems with large reaction networks.

1.3.5.3 Work by others: G-protein coupled receptors

The GPCRs constitute the largest family of transmembrane receptors, consisting of 5 subfamilies (Alberts, 2008; Rosenbaum et al, 2009). These proteins, whose structure and function were reviewed recently by Rosenbaum et al. (2009), are characterized by seven transmembrane spanning α-helical segments (Alberts, 2008; Fuxe & Kenakin, 2010). They regulate many physiological functions such as vision, gustation and olfaction (Rosenbaum et al, 2009; Vilardaga et al, 2010). Neurotransmitters, hormones and environmental stimuli activate these pathways. GPCRs are also implicated in many human diseases, such as inflammation, retinitis pigmentosa, nephrogenic diabetes insipdus, and Kaposi's sarcoma (Fuxe & Kenakin, 2010; Insel et al, 2007; Vilardaga et al, 2010; Waller et al, 2004). At present, most pharmaceutical drugs used by humans target GPCRs by serving as agonists or antagonists (Fallahi-Sichani & Linderman, 2009; Vilardaga et al, 2010).

Many aspects of GPCR signaling are well established. In the classical view, binding of ligand to a GPCR induces a conformational change in the receptor. The activated receptor initiates guanine nucleotide exchange (GDP→GTP) in its principal signaling partner, a heterotrimeric ($\alpha\beta\gamma$) G-protein complex. Like ras, heterotrimeric G proteins are tethered to the cytosolic leaflet of the plasma membrane through covalently attached lipids, and assume an active state once bound to GTP. An additional step is required for heterotrimeric G proteins: the separation of the GTP-bound Gα subunit from the G$\beta\gamma$ subunit, which diffuses into the cytosol. The subsequent activation of downstream effector proteins results in various distinct biological reactions.

Recent work has focused on new aspects of GPCR signaling, such as the evidence that at least some GPCRs can form homo- or hetero-dimers (Brinkerhoff et al, 2004; Fuxe & Kenakin, 2010; Waller et al, 2004). These dimers can interact further to form oligomers (Fallahi-Sichani & Linderman, 2009). Although believed essential for signaling to occur, the dimerization mechanism is well characterized for only a few GPCRs (Lambert, 2010). Due to the importance of GPCR signaling in healthy and diseased states, GPCR interactions, along with membrane organization, and their impact on signaling must be well characterized. Mathematical modeling is therefore being used increasingly to help unravel the intricacies of this pathway. A useful review of mathematical models that have been developed to study GPCR signaling is given by Linderman (2009).

Brinkerhoff et al. (2004) used triangular lattice-based Monte Carlo (MC) models to simulate receptor dimerization and activation in a two-dimensional plane, examining how dimerization creates clusters of receptors. Their model demonstrates the applicability of MC methods to systems with discrete reactions that are diffusion limited (Brinkerhoff et

al, 2004). Randomly selected particles undergo either one of two possibilities at each time step: displacement in a random direction by a distance governed by the diffusion coefficient or a chemical reaction. Reaction possibilities considered were receptor dimerization, binding of ligand by receptor, receptor activation of G protein, and receptor phosphorylation. This group's simulations suggest that clustering arises through both dimerization and cross talk between receptors as they approach one another closely and are able to share an effector. They also concluded that the resulting clustering enhances signaling.

Fallahi-Sichani et al. (2009) investigated lipid raft impact on GPCR signaling with a combination of MC (stochastic) and deterministic models. A lattice-based, kinetic MC model was used to establish the effects of low-diffusivity rafts on receptor dimerization and cluster dynamics. The stochasticity of the model allowed for receptor distributions to be examined, leading to parameter estimations for exploring the effects on downstream signaling using an ODE model. The fraction of plasma membrane covered by microdomains (rafts), which was varied from 2-30%, had a significant impact on output. At 2% coverage, microdomains amplified the overall response, but at higher coverage the signal was attenuated. They concluded that dimerization and lipid raft trapping cooperatively control the extent and dynamics of GPCR signaling.

Tolle et al. (2010b) developed an off-lattice, Brownian diffusion-based stochastic model, which they used to determine how AMPAR (alpha-amino-3-hydroxyl-5-methyl-4-isoxazolepropionic acid receptor) diffusion in the dendritic spine affects synaptic signaling, specifically Long-Term Potentiation (LTP).(Tolle & Le Novere, 2010a) LTP, an increase in synaptic strength, is a well-studied form of synaptic plasticity, the ability to

change the strength of a signal (Santamaria et al, 2010; Tolle & Le Novere, 2010a). Tolle et al.'s model accounts for the dendritic spine membrane, membrane receptors and scaffolding proteins known to bind to membrane receptors (Tolle & Le Novere, 2010a). The spatial domain representing the plasma membrane of the synaptic spine was modeled as a square of surface area corresponding to the measured volume of the spine. This square was separated into two different compartments or domains, in order to account for the two physiologically different portions of the plasma membrane: the post-synaptic density (PSD) and the extra-synaptic membrane (ESM). The PSD is a protein-rich region where AMPARs are concentrated (Santamaria et al, 2010; Tolle & Le Novere, 2010a), while the rest of the membrane is classified as the ESM (Tolle & Le Novere, 2010a), The transmembrane receptor movement within the ESM was modeled with Brownian-type diffusion, while confined motion was used to model the restricted diffusion within the PSD. Simulation results indicate that randomly placed receptors quickly localize to the PSD, which Tolle et al. (2010a) suggest explains the quick onset of LTP.

1.3.6 Concluding remarks.

This review specifically considers the mathematical modeling of protein clustering on the plasma membrane and the evidence that signal transduction can be enhanced by locally high concentrations of proteins that increase the probability of protein-protein interactions. This feature is especially important when the numbers of particles are small. When proteins are overexpressed, as in EGFR amplication in certain cancers, clustering may not be as significant (Hsieh et al, 2008). The role of membrane microdomains in signaling may be quite complex, since both inhibitory and stimulatory effects have been

observed experimentally and theoretically (Allen et al, 2007; Costa et al, 2009a; Miura et al, 2001; Pike, 2003).

Mathematical modeling, in conjunction with biological experiments, is providing new insights into the mechanisms that govern protein clustering in membranes and the resulting impact on signaling. Increasing experimental detail is being matched by increasingly complex models that account for previously ignored biological subtleties (Chakraborty et al, 2003; Erban & Chapman, 2009; Grima & Schnell, 2008; Li et al, 2008; Radhakrishnan et al, 2010; Resat et al, 2009; Turner et al, 2004). An important goal is to predict the functional responses of whole cells and cell-tissue systems, based upon integration of spatial and temporally encoded signals from surface receptors. Achieving this goal will necessitate the development of efficient and accurate multi-scale simulation capabilities. A daunting challenge to mathematical modeling of cell signaling continues to be the scaling up of computationally intense methods developed for studying molecular behavior to enable predictive modeling at progressively more complex levels, from the cellular to the systemic.

1.3.7 Acknowledgements

This work was supported by NIH R01CA119323 (to BW), NIH P50GM085273 (the New Mexico Spatiotemporal Modeling Center) and NIH K25CA131558 (to AH).

1.4 RECEPTOR TYROSINE KINASES: THE ERBB FAMILY

1.4.1 ErbB1

ErbB1 is part of the ErbB family of receptor tyrosine kinases (RTK) (Citri & Yarden, 2006). ErbB1 is not only critical for normal growth and development, but has also been implicated in various forms of cancer (Yarden & Sliwkowski, 2001). The ErbB1 receptor

fluctuates between a closed and open conformation (Baselga & Swain, 2009), with the open conformation being stabilized through ligand binding. When the receptor is in the open conformation, the dimerization arm is exposed, allowing the receptor to form homodimers with other ErbB1 receptors and hetero-dimers with the other three members of the family, ErbB2, ErbB3, and ErbB4 (Yarden & Sliwkowski, 2001). Due to the fluctuation of receptor conformation, dimerization between receptors is possible in the absence of ligand, forming pre-formed dimers (Schlessinger, 2002). The ability of non-ligand bound ErbB1 monomers to partner with one another, as well as with ligand-bound monomers, leads to multiple ligand:receptor dimer configurations.

When the receptors form a dimer, the kinase tails form an asymmetric configuration connecting the C-lobe of one kinase tail to the N-lobe of the opposite kinase tail. This conformation is often referred to as an asymmetric dimer (Zhang et al, 2006). When the kinase tails interact, the tails are phosphorylated through transphosphorylation. This phosphorylation, sometimes referred to as dimer activation, triggers a signal cascade, recruiting different effector and adaptor proteins to the tail (Linggi & Carpenter, 2006; Yarden & Sliwkowski, 2001). Macdonald-Obermann et al.'s (2009) study showed an alternate view of transphosphorylation in light of the asymmetric dimer configuration. In their phosphorylation mechanism, the C-lobe of one receptor, deemed the "activator", activates the N-lobe of the opposite tail, deemed the "receiver", which then phosphorylates the tail of the activator.

Recent SPT studies published by Low-Nam et al. (2011) give even greater insight into the kinetics and dynamics of ErbB1. Using a Hidden Markov Model to analyze the SPT data, they were able to separate receptor diffusion in to three distinct states, free,

confined, and dimer. This type of resolution allows for more accurate diffusion coefficients to be derived from the data. Low-Nam et al. (2011) found that ErbB1 dimers diffuse much slower than ErbB1 monomers. Further analysis of the SPT data revealed the interaction dynamics between receptor pairs. They were able to calculate dimer off rates by studying the characteristics of the distance between two receptors over time. This analysis gave insight into the kinetics of the different dimers (preformed, 2 ligand:2 receptor, and 1 ligand :2 receptor), showing that ligand does in fact stabilize the dimer, leading to lower off rates. Finally, studying the distance between receptor pairs lead to the observation that a pair of receptors will interact multiple times throughout the course of a minute, forming dimers and falling apart again.

1.4.2 ErbB2 and ErbB3

ErbB2 is different from the other 3 members of the ErbB family due to its structure. While ErbB1/3/4 all are typically in a closed conformation when no ligand is bound, ErbB2 is always in the open conformation without the presence of ligand. There is no known ligand associated with the ErbB2 receptor (Baselga & Swain, 2009; Linggi & Carpenter, 2006). Overexpression of ErbB2 is rampant in breast cancer, making ErbB2 a typical target for therapeutics (Baselga & Swain, 2009). ErbB3 has recently become a major focus of study for the ErbB family (Baselga & Swain, 2009). Studies show that overexpression of ErbB3 is also linked to breast cancer (Yarden & Sliwkowski, 2001), and ErbB3 is now being studied as a target for possible therapies (Baselga & Swain, 2009). Together, ErbB2 and ErbB3 are known to form oncogenic dimers, with extremely potent signal cascades (Zhang et al, 2012).

Published works by top ErbB researchers have suggested that the ErbB3 receptor cannot form a homodimer (Baselga & Swain, 2009; Berger et al, 2004). However, recent SPT experiments published by Steinkamp et al (2014) have suggested that ErbB3 does in fact form homodimers, extremely stable homodimers. SPT experiments have shown that ErbB2 and ErbB3 form dimers, though they are shorter lived than the ErbB3 homodimers. Kinase assay experiments by Steinkamp et al. (2014) have also given insight into the activation state of ErbB3, showing that there is measurable kinase activity in ErbB3 in the presence of heregulin. The current accepted signaling unit is ErbB2-ErbB3 (Baselga & Swain, 2009; Linggi & Carpenter, 2006; Warren & Landgraf, 2006; Yarden & Sliwkowski, 2001), however, the results seen from the SPT experiments, as well as kinase assays by Steinkamp et al. (2014), suggest that the main signaling unit may in fact be the ErbB3 homodimers. A possible activation mechanism would be that ErbB2 must first activate ErbB3's kinase domain through dimerization, then the active ErbB3 receptor would be able to form a dimer with, as well as activate, another ErbB3 receptor and cause the mitogenic signal cascade.

1.5 RESEARCH FOCUS

The focus of this dissertation work is the ErbB family of receptor tyrosine kinases. Specifically, we look at the kinetics and dynamics of these receptors on the cellular membrane through a spatial stochastic model. Recent advancements in experiments, mainly in single particle tracking, have been a strong motivation for this work. Many exciting and interesting observations have been made through SPT, and we knew integrating those results and observations with a spatial stochastic model would enable us

to look even deeper into the meaning and implications of those results. The methods used to create our spatial stochastic model are discussed in **Chapter 2**.

We started out by focusing on the homo-interactions of ErbB1 to build our model, validate it, and then predict how the observed dynamics are impacted by different factors. We proposed that the repeated interactions between ErbB1 receptors, as observed through SPT, are a product of the membrane landscape and receptor density. We further suggested that the repeated interactions are a mechanism to facilitate phosphorylation through an asymmetric phosphorylation mechanism. Finally, we were able to investigate how these factors come together and impact the receptor's activation state over time. This work is discussed in detail in **Chapter 3**.

We then spent time analyzing the diffusion data from SPT experiments to get insight into the membrane landscape and it's role in receptor diffusion. From our previous work and the SPT experiments, we knew that confinement zones on the membrane impacted receptor diffusion. We sought to quantitatively understand the observed anomalous diffusion by studying the diffusion properties of proteins in the cellular membrane, and how different structures on the membrane contribute to these properties. This work is discussed in **Chapter 4**. We then took the diffusion analysis one step further to reconstruct the confinement zones encountered on the membrane during SPT experiments. We proposed we could estimate the size and shape of the confinement zones by converting the dynamic trajectory data into static spatial data and applying a clustering algorithm. The "Domain Reconstruction" Algorithm (DRA) is discussed in **Chapter 2** and implemented in **Chapter 5**.

Finally, we extended our spatial stochastic model to a heterogeneous population to investigate the dynamics between ErbB2 and ErbB3, and used the DRA to create our simulation space. Our focus was to investigate the interactions and phosphorylation kinetics between ErbB2 and ErbB3. In this work, we truly integrated experimental work and modeling. We set out to investigate the impact of ErbB3 activation on the currently accepted ErbB2-ErbB3 interaction model and how the membrane landscape impacts those interactions. Our final goal was to then use our ErbB2-ErbB3 spatial stochastic model to study possible mutations in ErbB3 and see impact the activation states of ErbB2 and ErbB3. This work is discussed in **Chapter 5**. In **Chapter 6**, we bring all the studies together and discuss their significance and impact in the fields of membrane biology and ErbB receptors.

CHAPTER 2: MODELING METHODS

2.1 OVERVIEW

The use of mathematical models to describe and investigate various biological systems has been increasing exponentially over the past decade. In response to this, many modeling methods have either been applied from other scientific areas or been developed out of necessity to describe new behaviors (Andrews et al, 2010). These models range from population style deterministic models all the way to intricate single particle stochastic models. The system being studied, as well as the questions to be answered, dictate the necessary type of model and the level of resolution that model will need (Andrews et al, 2010). As experimental methods have been developed over time, the resolution of these experiments has also increased. The uptick in single molecule experiments and the resulting detailed data call for modeling methods to be further developed.

Here we describe methods for generating a model with spatial resolution and single molecule precision utilizing single molecule experimental data. Our approach takes a logical progression from initial biological system design through simulation. Important spatial aspects must first be determined along with the reaction network of the system. Once the model system is well defined, the dynamics and kinetics can be derived and the necessary parameters obtained. After the model is completely developed, the relevant outputs of the model needed to test the motivating hypothesis can be defined and the model can then be implemented.

2.2 SIMULATION SPACE CHARACTERIZATION

2.2.1 Summary

Introduction of a two dimensional space into a model requires a few extra specifications above and beyond those in non-spatial models. The minimum requirements to define this surface are the spatial limits and the initial distribution of proteins on the surface. Additional details may also be specified, such as membrane obstacles, keeping in mind as more and more detail is added to the 2D surface, the expense (time) of the simulation increases.

Generation of the simulation space is based on available physiological information about the model system, as well as restrictions of the simulation. Images obtained through experiments providing single particle information, such as an electron micrograph, can be used to import an initial distribution of the protein of interest. Similarly, single particle tracking data can be used to reconstruct confinement zones receptors encounter on the membrane. Combining this data with information about the receptor population allows for a good estimate for a simulation space and initial conditions. Here we discuss in detail the two methods we have developed, an Image Importer and Domain Reconstruction Algorithm (DRA), to import and analyze this data to recreate physiologically relevant simulation spaces.

2.2.2 Image Importer

2.2.2.1 Motivation

One method used to visualize ErbB receptors on the cellular membrane is electron micrography (EM). The receptors are fixed on the cellular membrane, labeled with immuno-gold particles, and then imaged (Yang et al, 2007). The resulting EM images

give insight into receptor distribution on the membrane. ErbB1 receptors tend to cluster on the membrane. These pockets are often referred to as confinement zones or domains. This view of receptor distribution on the membrane is one of the observations that lead to the addition of spatial resolution to our model. In a first step towards integrating experimentally observed receptor distributions for initial spatial conditions, we developed a method to import the receptor positions from EM images and estimate domain sizes based on the receptor clusters.

2.2.2.2 Particle Import and Domain Estimation

The EM image importer uses the Image Processing Toolbox in Matlab. The program has been setup to walk the user through importing data through various popup instruction boxes. The programs algorithm depicted as a flowchart is shown in Figure 2.1. An example screenshot of using the importer is shown Figure 2.2. This algorithm is implemented in **Chapter 3.**

EM Image Importer Function

1. Specified image is read in to Matlab.
2. User specifies if domains will be included, and if so, how many.
3. Scale bar length is read in. This will give a unit conversion between pixels and the physical units the image is in. The units are specified within the .m file.
4. x and y limits are read in. These are the absolute boundaries for the simulation space.
5. If domains are to be included, they are read in one at a time until the number of specified domains in Step 2 is met.

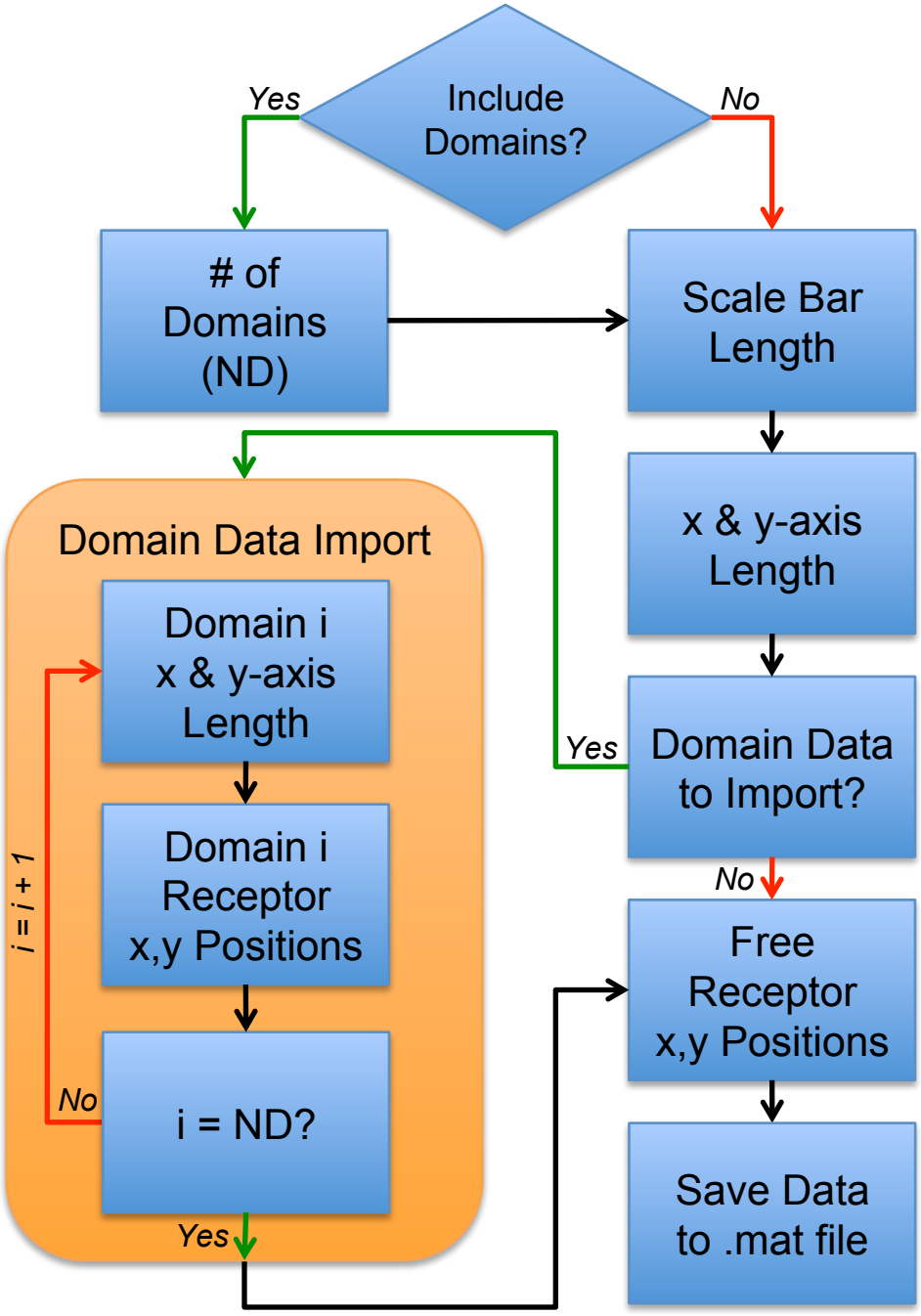

Figure 2.1: EM Image Importer Algorithm Flowchart.

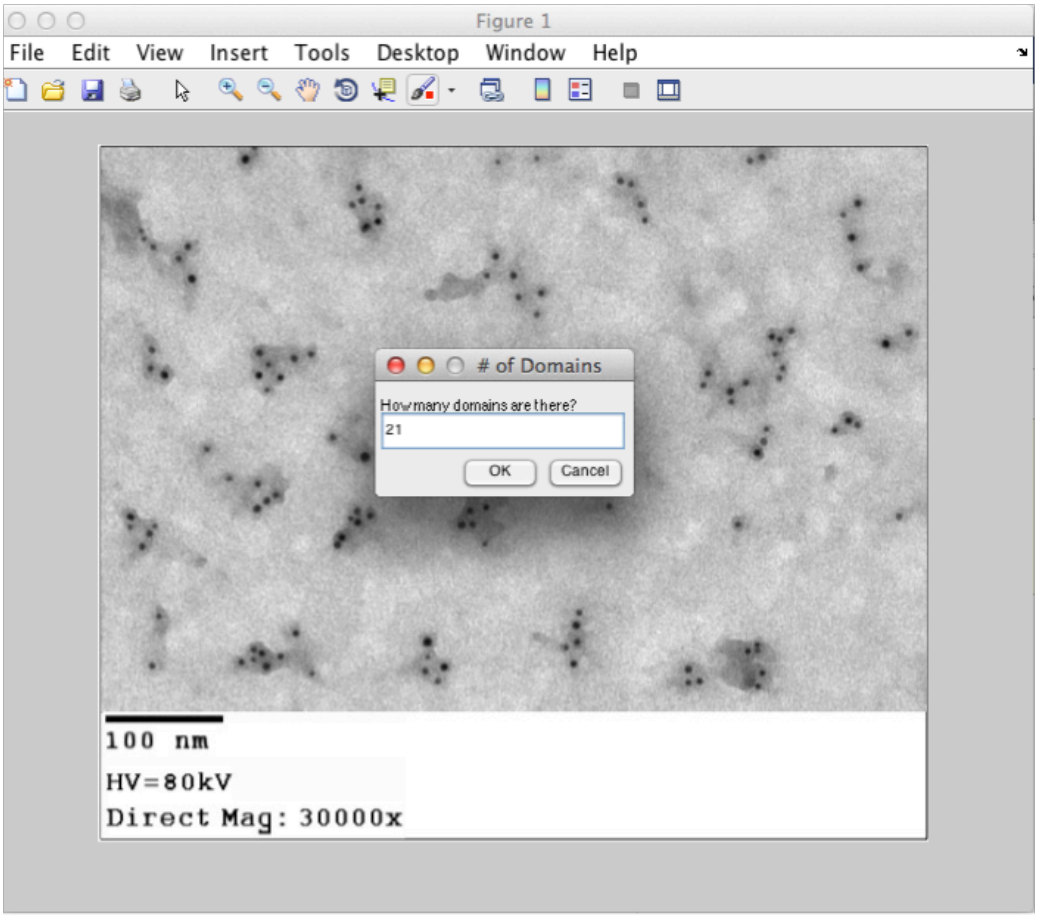

Figure 2.2: EM Image Importer Screen Shot. The EM Image Importer walks the user through importing receptor and domain positions through a series of prompts.

a. Domain x and y limits are read in. These limits are defined by the cluster size of receptors in the image. The domain is a rectangle defined by the x and y limits.
b. Receptor x,y positions within the specified domain are read in.

6. Receptor x,y positions not in domains are read in.
7. The total number of receptors is calculated and all the data collected is saved to a .mat file for later use.

2.2.3 Domain Reconstruction Algorithm

2.2.3.1 Motivation

While the necessity of spatial resolution has been shown and modeling methods have been developed to address it, characterization of the simulation space using single particle data has not yet been explored. Here we take the addition of spatial resolution one step further by directly incorporating single particle tracking data to describe and represent the simulation space features and particle distributions. Many studies involving single particle tracking have observed a particle diffusion pattern that suggests the particles are in a confined area (Kusumi et al, 1993; Low-Nam et al, 2011; Saxton, 1993). Taking into account this observation, we have created an algorithm that analyzes single particle tracking data to reconstruct the size and shape of these confining areas. A flow diagram of the algorithm is shown in Figure 2.3. The algorithm first picks out the points in the trajectories determined to be in a confined space, and then performs a cluster analysis on the points. The clusters are then converted into contours, which are then "inflated" to reconstruct the shape of the confining space. This algorithm is implemented in **Chapter 5**.

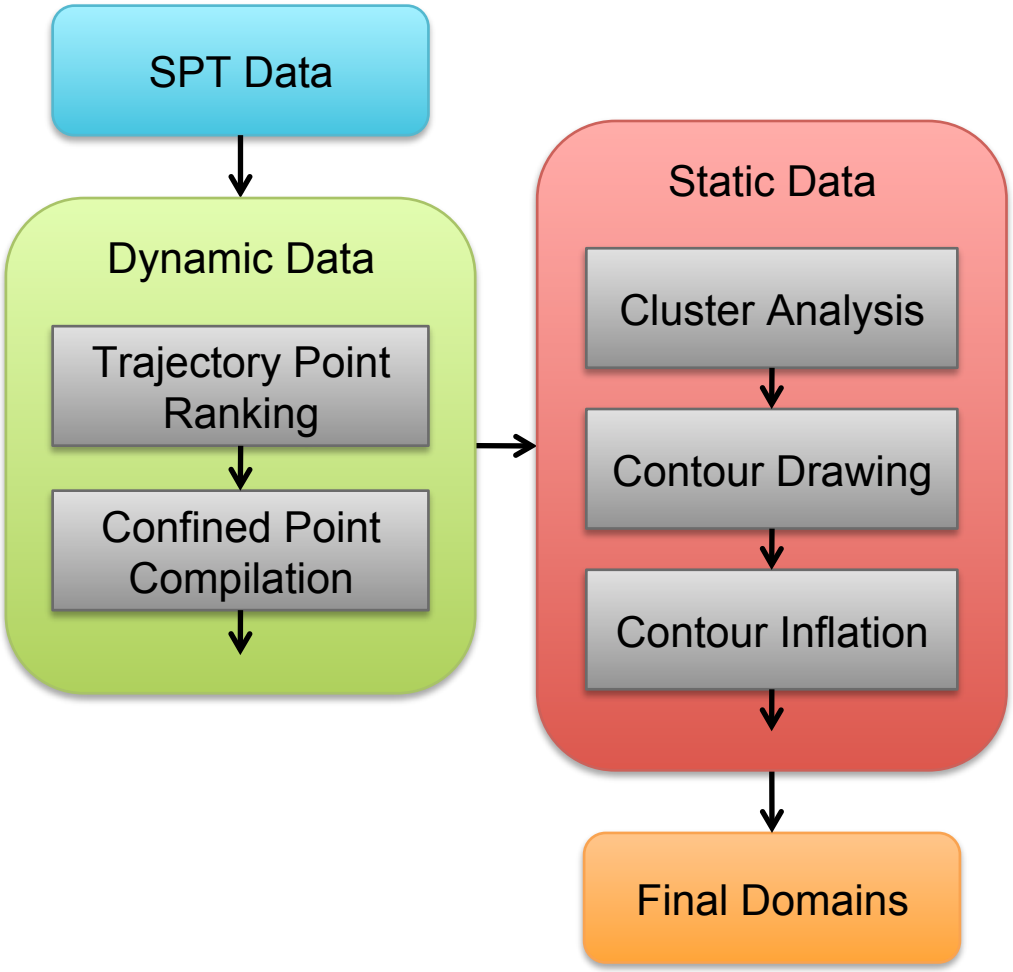

Figure 2.3: Domain Reconstruction Algorithm Flowchart. The DRA starts from dynamics single particle tracking data. The algorithm ranks the data and compiles the points. The data is then converted to static spatial data and used to determine domain size and distribution. The domains are reconstructed via a clustering algorithm, contour drawing, and contour inflation.

2.2.3.2 Trajectory Analysis

One of the major undertakings of this algorithm is the need to transform dynamic trajectory data into static spatial data. To accomplish this task, we devised a ranking system that sorts every point in an individual trajectory into two groups: confined or free. The confined points are what would be considered to be in a confinement zone, while the free points are considered to be "free" on the membrane.

To rank these points, we first calculate the jump size (displacement) for a selected number of time steps (1, 2, 5, 10, 20, 50, 100, 200). The time step is applied moving forward through the trajectory as well as backwards through the trajectory. Covering this range of step sizes accounts for "holes" that may be in the trajectory data due to experimental conditions such as blinking of the quantum dots or quantum dots temporarily moving out of the focal plane. These jump sizes are calculated for each trajectory in a set of comparable SPT files, then aggregated by step size.

Once the jump sizes have been compiled, the individual points from each trajectory are ranked. For every point, the relative rank is calculated by comparing the point's jump size to all the other points' jump sizes for a specific step size. The jump sizes for a specific step size are sorted in order from smallest to largest. The rank of a specific point is where it falls in that order. This is repeated across all points for each step size.

Some points may not have a score for all 16 step sizes due to the aforementioned holes in the trajectories. To account for this, a weighted average is used to determine the overall rank for each point. First, for each point, the forward and backward ranks for a step size are combined. The combined ranks for each step for that point are then averaged together; this is the final rank for that point.

2.2.3.3 Cluster Analysis

We rely on distance based hierarchic clustering. This approach had been widely used in the literature, ranging from ecology and genomics to receptors on the cell membrane (Espinoza et al, 2012). Our method is a modified version of that developed by Espinoza and coworkers for TEM images of receptors labeled with gold nano-particles.

Given N points $\{P_1,P_2,\ldots,P_N\}$ in a plane, with coordinates $\{(x_1,y_1),\ldots(x_N,y_N)\}$, we want to partition them into mutually exclusive groups or clusters in a way that reflects their proximity or similarity to each other. This is not a clearly defined notion and the appropriate clustering method should be ultimately determined by the experimental context. Here, we use the "slow points" identified from jump size distributions as indicators of an underlying physical structure (such as lipid rafts); therefore the notion of proximity defined by physical distance to the closest members of the cluster is more appropriate considering, for instance, an average distance to the entire cluster (this is the idea underlying K-means clustering).

We construct clusters by comparing the distance $d(P_1,P_2)=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$ between points to a reference length L, sometimes called a "scale". Two points A, B are in the same cluster if their distance $d(A,B)\le L$; we denote this relationship by $A \sim_L B$. We extend the relationship by transitivity: if $A \sim_L B$ and $A \sim_L C$, then $B \sim_L C$. It is easy to see that the procedure will induce a partition of the set of points. (The $\sim_L$ relation is symmetric and transitive, therefore it is an equivalence relation in the mathematical sense; the clusters are the corresponding equivalence classes.)

The partition into clusters is unique for a given set of points and length scale L. Any two points in a cluster can be linked by a connected path consisting of line segments of length $\leq L$ connecting points from the same cluster (Figure 2.4A).

2.2.3.4 Contour Drawing

In order to build a geometric area (shape or footprint) around a given cluster, we start with the union of the circles of radius *L/2* centered on all members of the cluster. We assume the cluster resulted from hierarchic clustering with distance parameter *L,* so any member of the cluster must be reached from any other member through a sequence of segments connecting cluster points, such that no individual segment is long than L. The connection graph in Figure 2.4 constructed by putting an edge between all pairs of points whose distance is $\leq L$ must be connected, and therefore, the union of the circles (Figure 2.4B) must form a contiguous area.

To straighten the boundary of the region defined this way, we extend the area by adding rectangles of height L along the contour of the connection graph (double lines Figure 2.4C). The reconstructed region is the reunion of the inside of the contour graph, and the circles and rectangles around the vertices and edges of the contour (shaded, respectively grey areas on Figure 2.4D).

The contour graph or "tight contour" for a cluster of points is defined by the sequence of boundary points and the segments that connect them. The list is constructed by adding new points to the contour, based on the existing points and a reference direction. As the contour is built, it circles around the points in a counter-clockwise direction, so that all the interior points are to the left hand side. When the process is finished, the last point in the contour is identical to the first one.

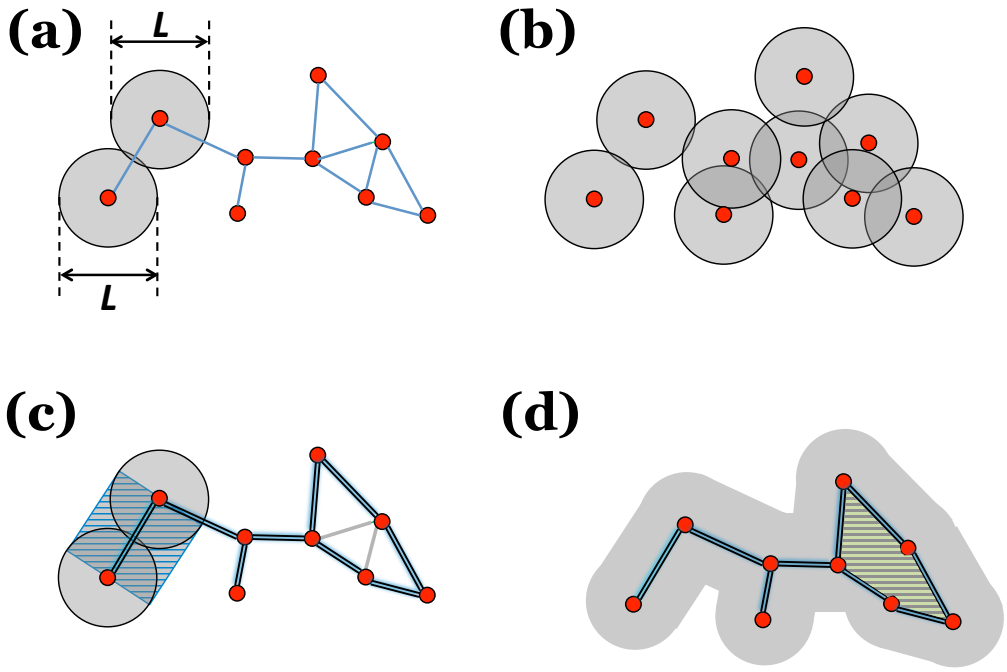

Figure 2.4: Clustering Algorithm Walkthrough. (a) The points in a cluster form a connected graph, where edges connect points whose distance is less than the length scale L. Circles of diameter L centered on two points intersect if and only if the points are connected in the sense described above. (b) We want to define the footprint based on the reunion of all the circles of diameter L, centered on the points in the cluster. (c) We first identify the outer contour (sometimes a skeleton, with no interior) of the cluster graph (double blue shaded lines). We 'pad' the area defined by the circles by adding rectangles along the edges of the contour graph. (d) The reconstructed region is the reunion of the inside of the contour graph (if any), and the circles and padding rectangles around the vertices and edges of the contour graph.

Contour Building Algorithm

1. Start with the rightmost point of the group; set the reference direction pointing to the right. (Any point on the convex hull is acceptable; the reference direction needs to be pointing toward the outside of the hull.)
2. Add new points:
    a. Identify all the points in a circle of radius *L* centered on the last point added (the current point); these are the candidate points.
    b. Draw line segments from the current point to each candidate point. If this intersects a segment in the already identified part of the contour, discard respective candidate point.
    c. Order the remaining candidates by the clockwise angle from the reference direction to the segment connecting the current point to the candidate; choose the candidate with the smallest angle and add it to the list.
    d. Set reference direction to point from the newly added point back to the previously added point
3. The process terminates when the same *segment* is added to the contour. The same point may be visited twice, in opposite directions. Upon successful termination, the last point in the list is the same as the first one.

NOTE: The contour defined this way is not unique, but the algorithm always returns a contour that is a refinement of the convex hull of the points, is not self-intersecting, and contains no edges longer than *L*.

2.2.3.5 Contour Inflation

The tight contour or contour graph defines the reconstructed region. This contour is often lacking an "inside", and can have sections that are just a single chain of points. The rationale of padding is to represent the area "of influence" of each point in the cluster – roughly, the set of geometric locations that are closer to this cluster than to any points that are not part of the cluster.

The padding adds two types of elements to the core contour and its interior:

1. A rectangle of height $L/2$ on the outer side of each edge of the tight contour graph
2. A sector of a circle of radius $L/2$ at each vertex with a positive (convex) angle

The contour inflation algorithm constructs a second contour (the outer or padded contour) that encloses the first one. Similarly to the tight contour, the padded contour defined as a polygonal line; the vertices of the polyline are all auxiliary points, and are not elements of the cluster.

The contour inflation algorithm proceeds along the inner contour, and builds the outer contour parallel to it, adding one or more points to account for each vertex of the inner contour. If the angle at the respective vertex is positive (convex), we add the corners of the padding rectangles and points on the arc of the padding circle centered on the vertex; if the angle is negative (concave), then we only add one point, namely the intersection of the two padding rectangles.

Below we describe the algorithm we employed. This algorithm also relies on a reference direction, which corresponds to the segment preceding the current vertex (with the direction defined by that of the algorithm – counterclockwise in our case).

Contour Inflation Algorithm

1. Start at a point on the tight contour. Moving counter-clockwise along the contour, set the reference direction to that of the incoming edge in the counter-clockwise direction, and the current direction to the outgoing edge.
2. The elements of the new contour that are added to represent the current vertex are set based on the angle between the incoming and outgoing directions.
    a. If the angle is positive (left turn, convex vertex), then add a sector of a circle of radius $L/2$, centered on the current vertex, and delimited by the radii perpendicular to the incoming and the outgoing edges. The sector is discretized as a sequence of points starting with the end the radius orthogonal to the incoming edge and ending at the other one. At least one intermediate point is added at the end of the radius that is the symmetry axis of the sector.
    b. If the angle is negative (right turn, concave vertex), we only add the point where the two adjacent rectangles intersect. This point is also on the symmetry axis (bisector) of the angle on the inner contour.
3. The algorithm proceeds along the vertices of the inner contour, adding points to the outer contour for each of them. The process terminates when the starting vertex is reached, and is about to be passed in the same direction as in the first iteration.

2.3 SPATIAL STOCHASTIC MODEL

2.3.1 Motivation

Any of the stochastic modeling methods mentioned previously may be implemented with the simulation space derived above. For our work, we chose Smoluchowski dynamics with modifications proposed by Andrews and Bray (2004). We felt that this model most accurately represented physiological processes of the known available models. We opted to develop our own version of Andrews and Bray's (2004) model, instead of directly using their simulator SMOLDYN, for two distinct reasons:

1. Inclusion of our reconstructed simulation spaces, mentioned above.
2. Unique Receptors. We needed to be able to keep detailed records about each receptor over the simulation time.

While SMOLDYN is a very effective and efficient simulator, both of the above mentioned capabilities were not available in SMOLDYN at the time we started this work. We developed the code in FORTRAN 90 for ease of use and the ability to be compiled. Here we will go through the steps necessary to define the model as well as how to implement the experimental data.

2.3.2 Biological Reaction Network Definition

Deriving the reaction network for a system of interest is a very important step. Leaving out a reaction or over specifying the system can lead to incorrect conclusions. We suggest a bottom up approach, starting as simple as possible and building on to the network, as necessary. Our reaction network includes a series of first order reactions and second order reactions.

1. First Order Reactions

a. Dimer Dissociation

b. Phosphorylation

c. Dephosphorylation

d. Domain Escape (Confined Receptor → Free Receptor)

2. Second Order Reactions

a. Receptor dimerization

2.3.3 Particle Diffusion

Once the simulation space and reaction network have been specified, the diffusion of the proteins and the reaction kinetics must be accounted for. Receptor diffusion is based on Brownian motion. Brownian motion is represented in a simulation by picking a random number from a normal distribution and applying that value to the root mean square (RMS) step length, recreating the stochastic nature of diffusion (Andrews & Bray, 2004; Kusumi et al, 1993; Popov & Agmon, 2001).

$$x(t+\Delta t) = x(t) + RMS\xi_x$$
$$y(t+\Delta t) = y(t) + RMS\xi_y$$
$$RMS = \sqrt{2D\Delta t}$$

where x and y are the receptor's Cartesian coordinates, RMS is the root mean square step, Δt is the time step, and ξ_x and ξ_y are the normally distributed random numbers. Periodic boundary conditions are used as a receptor approaches the edge of the simulation space.

Membrane obstacles create a hurdle for the diffusing receptors during the simulation. The diffusion depends on the type of membrane obstacle encountered. If the obstacle is simply a boundary, the receptor may either "jump" the obstacle or be reflected off the

boundary in the opposite direction. If the obstacle is a confining area, sometimes referred to as a sticky box or domain, the receptor typically has an easier time entering the confinement zone, however needs to pay a higher "toll" to escape. The toll is applied using a probability calculation using the transition rate, or escape rate, from one zone to the next:

$$P(Transition) = 1 - \exp(-r_{Transition}\Delta t)$$

However, this probability has been simplified further to:

$$P(Transition) = r_{Transition}\Delta t$$

due to the small time step. If a confined receptor is set to diffuse out of a confinement zone, and this escape probability is not met, the confinement zone assumes a reflective boundary condition, trapping the receptor inside.

2.3.3.1 Boundary Conditions

There are two sets of boundary conditions encountered during receptor diffusion, periodic and reflective. A periodic boundary is used when a receptor reaches the edge of the simulation space. As the receptor approaches the edge of the simulation space, the jump that takes the receptor across the boundary is split between the remaining distance before the boundary and the rest of the jump distance. The jump distance beyond the boundary is then transposed to the opposite boundary edge and continued back into the simulation space. Conditions and equations for implementing the periodic boundary condition are in Table 1.

	Minimum Bound	Maximum Bound
Cross y Boundary	$x_{Final} = x_{Initial} + \Delta x + L_x^{SimSpace}$	$x_{Final} = x_{Initial} + \Delta x - L_x^{SimSpace}$
Cross x Boundary	$y_{Final} = y_{Initial} + \Delta y + L_y^{SimSpace}$	$y_{Final} = y_{Initial} + \Delta y - L_y^{SimSpace}$

Table 1: Periodic Boundary Conditions. Equations for calculating a receptor's new position when a simulation space boundary is crossed.

Reflective boundary conditions are used for simulation space confinement zones. When a receptor encounters a confinement zone that it cannot cross, the receptor is directly reflected away from the obstacle. The receptor jump is split again between the distance to the obstacle and the distance the receptor would move beyond the obstacle. This second distance is the distance the receptor is moved away from the obstacle. Conditions and equations for implementing the reflective boundary condition are in Table 2.

	Minimum Bound	Maximum Bound
Cross y Boundary	$x_{Final} = 2x_{Min}^{Obstacle} - \Delta x - x_{Initial}$	$x_{Final} = 2x_{Max}^{Obstacle} - \Delta x - x_{Initial}$
Cross x Boundary	$y_{Final} = 2y_{Min}^{Obstacle} - \Delta y - y_{Initial}$	$y_{Final} = 2y_{Max}^{Obstacle} - \Delta y - y_{Initial}$

Table 2: Reflective Boundary Conditions. Equations for calculating a receptor's new position when it encounters an obstacle in the simulation space.

2.3.4 Receptor Kinetics

Modified Smoluchowski dynamics are used to simulate receptor kinetics (Andrews & Bray, 2004). The simulator focuses on one receptor at a time, picking every receptor once, on average, over a fixed time step. Receptors are picked randomly using a uniform distribution, allowing the order the receptors are moved or reacted to be different for each

time step. Once a receptor is picked, the possible reactions for that receptor are examined and applied as appropriate. There are two types of reactions possible, 1st order and 2nd order.

First-order reactions are implemented through a probability calculated using the reaction rate and simulation time step:

$$P(reaction) = 1 - \exp(-r_{reaction}\Delta t)$$

This probability has been simplified further, as in Hsieh et al. (2008) and Andrews and Bray(2004):

$$P(reaction) = r_{reaction}\Delta t$$

Second-order reactions require more detail to account for two receptors interacting with one another. A receptor's likelihood to react with another receptor is based on a distance termed the binding radius. The binding radius takes into account the dimer on rate, diffusion coefficient of the receptors that will comprise the dimer, and the simulation time step (Andrews & Bray, 2004). While the binding radius is not a physical radius relative to the size of the receptor, the use of this radius allows the simulation to be closer to the physical situation than previous methods using probabilities (Andrews & Bray, 2004; Erban & Chapman, 2009; Gillespie, 1977; Hsieh et al, 2008; Popov & Agmon, 2001). When the chosen receptor is moved, the final position is scanned for other receptors within the binding radius. If a receptor is within this distance, the receptors will react. Calculation of the binding radius is done through an iterative process where the experimentally determined reaction on rate is compared to the simulation's on

rate of reaction. This causes the binding radius to be a function of diffusion coefficients, the reaction on rates, and the simulation time step (Andrews & Bray, 2004).

When a reversible reaction occurs, an unbinding radius is used to set the dissociating receptors apart. The unbinding radius is calculated such that the occurrence of an unrealistic amount of repeated interactions is minimized:

$$\phi_s = \frac{\sigma_b}{\sigma_u}$$

where σ_b is the binding radius and σ_u is the unbinding radius. The default ratio of binding radius to unbinding radius is 0.2 (Andrews & Bray, 2004).

2.3.5 Parameters

The necessary parameters become apparent once the diffusion and reaction equations have been defined. Ideally, the parameters would come from relevant experimental data with single particle precision. This is currently not possible for many systems; therefore the next best source would be ensemble data from a relevant experimental system. If experimental data is not available for a specific parameter, either through experiments or literature, a last resort method would be to fit the parameter. Zhang et al. (2009) showed the importance of where parameters come from, and show that mixing parameters from different cell lines can cause incorrect conclusions to be drawn from simulations. For this reason, we have tried our best to obtain parameters that have all originated from the same cell lines relevant to the receptors focused on in each of our models. Our parameters mainly come from SPT experiments, however kinase assays were used for the phosphorylation/dephosphorylation rates. The specific parameter sets and where they came from are discussed in each chapter that implements this modeling scheme.

2.4 NOTES

Matlab scripts for the Simulation Space sections are available in **Appendix A**. Fortran code for the spatial stochastic model is available in **Appendix B**.

2.5 ACKNOWLEDGEMENTS

Meghan M. Pryor developed the EM Image Importer. Meghan M. Pryor, Adam M. Halasz, and Jeremy S. Edwards developed the Domain Reconstruction Algorithm and Spatial Stochastic Model code. Michael Wester offered invaluable programming advice during the development of both the Domain Reconstruction Algorithm and Spatial Stochastic Model. Shalini T. Low-Nam, Mara Steinkamp, Diane S. Lidke, and Bridget S. Wilson contributed to the reaction network development, biology considerations, and parameter compilation.

CHAPTER 3: DYNAMIC TRANSITION STATES OF ERBB1 PHOSPHORYLATION PREDICTED BY SPATIAL-STOCHASTIC MODELING

Meghan McCabe Pryor,[†] Shalini T. Low-Nam,[§#] Ádám M. Halász,[‡] Diane S. Lidke,[§ ¶] Bridget S. Wilson,[§ ¶*] and Jeremy S. Edwards[† ¶ ‖*]

[†]Department of Chemical and Nuclear Engineering, University of New Mexico, Albuquerque, New Mexico; [§]Department of Pathology, University of New Mexico Health Sciences Center, Albuquerque, New Mexico; [‡]Department of Mathematics and Mary Babb Randolph Cancer Center, West Virginia University, Morgantown, West Virginia; [¶]Cancer Research and Treatment Center, University of New Mexico Health Sciences Center, Albuquerque, New Mexico; [‖]Molecular Genetics and Microbiology, University of New Mexico Health Sciences Center, Albuquerque, New Mexico.

[#]Present Address: Shalini T. Low-Nam's present address is Department of Chemistry and Biochemistry, South Dakota State University, Brookings, South Dakota.

*Correspondence: BWilson@salud.unm.edu, JSEdwards@salud.unm.edu

3.1 ABSTRACT

ErbB1 overexpression is strongly linked to carcinogenesis, motivating better understanding of erbB1 dimerization and activation. Recent single particle tracking data have provided improved measures of dimer lifetimes and strong evidence that transient receptor co-confinement promotes repeated interactions between erbB1 monomers. Here, spatial stochastic simulations explore the potential impact of these parameters on erbB1 phosphorylation kinetics. This rule-based mathematical model incorporates structural evidence for conformational flux of the erbB1 extracellular domains, as well as asymmetrical orientation of erbB1 cytoplasmic kinase domains during dimerization. The asymmetric dimer model considers the theoretical consequences of restricted transactivation of erbB1 receptors within a dimer, where the N-lobe of one monomer docks with the C-lobe of the second monomer and triggers its catalytic activity. The dynamic nature of erbB1 phosphorylation state is shown by monitoring activation states of individual monomers as they diffuse, bind and rebind after ligand addition. The model reveals the complex interplay between interacting liganded and non-liganded species and the influence of their distribution and abundance within features of the membrane landscape.

3.2 INTRODUCTION

ErbB1 (EGFR, Epidermal Growth Factor Receptor) is the canonical member of the erbB receptor family (Citri & Yarden, 2006) and a critical player in normal growth and development, as well as carcinogenesis (Citri & Yarden, 2006). ErbB1 signaling is initiated by ligand-induced homo- and hetero-dimerization that is mediated primarily by engagement of extracellular dimerization arms (Baselga & Swain, 2009). Structural

evidence also suggests that the erbB1 extracellular domain fluctuates between the closed and open conformation in the absence of ligand (Baselga & Swain, 2009), transiently exposing the erbB1 dimerization arm and permitting transient "pre-formed" dimers to occur (Schlessinger, 2002). We previously used spatial stochastic modeling to predict the impact of receptor density, through local receptor trapping in membrane domains or receptor overexpression, on the rate of pre-formed dimers (Hsieh et al, 2008). The ability of non-ligand bound erbB1 monomers to partner with each other and with ligand-bound monomers leads to a complex mix of dimer configurations. Once dimers form, the signal is propagated by activation of integral tyrosine kinase activity in the receptor cytoplasmic tail, trans-phosphorylation of tyrosine residues in receptor tails and recruitment of cytosolic signaling partners (Citri & Yarden, 2006). Both deterministic and stochastic mathematical models have been developed to consider the complexity of erbB1 signaling, with successive generations of erbB1 models building on ever richer data sets for binding kinetics, phosphorylation/dephosphorylation dynamics and adaptor recruitment (Blinov et al, 2006; Costa et al, 2009b; Hendriks et al, 2003; Hsieh et al, 2008; Kholodenko et al, 1999; Kleiman et al, 2011; Radhakrishnan, 2010; Sasagawa et al, 2005; Schoeberl et al, 2002).

Not yet considered in mathematical models is the asymmetrical docking and activation of erbB1 cytoplasmic kinase domains, which accompanies extracellular domain dimer formation (Lu et al, 2012; Macdonald-Obermann & Pike, 2009; Mi et al, 2011; Zhang et al, 2006). In an asymmetric dimer, the N-terminal lobe of one kinase domain in the dimer pair interacts with the C-lobe of the other (Zhang et al, 2006). Mutagenesis and biochemical studies support an unusual transactivation model, where

activation of catalytic activity is restricted to the monomer whose C-lobe has been engaged. Thus, one monomer in the dimer pair is considered to be the "receiver" and the monomer contributing the N-lobe is considered to be the "activator". A novel aspect of the present study is the consideration of restrictions that asymmetrical docking theoretically imposes upon ErbB trans-phosphorylation into the spatial stochastic model, taking advantage of the flexibility of the model's rule-based framework.

Our model also builds on improved measures of erbB1 diffusive behavior and dimerization kinetics, made possible through remarkable advances in single particle tracking (SPT) methodology (Low-Nam et al, 2011). This recent study by Low-Nam et al. (Low-Nam et al, 2011) provided important new parameters for the spatial stochastic model. Among these values are the differential lifetimes of dimer pairs, based upon the occupancy of the ligand-binding site in each monomer. For example, the authors showed that dimer pairs comprised of two ligand-bound monomers have the longest lifetimes, compared to lifetimes of pairs comprised of one ligand-bound and one unliganded monomer or 2 unliganded monomers (Low-Nam et al, 2011). In addition, data from SPT experiments provided strong evidence for repeated interactions between two receptors while co-confined in specialized features of the plasma membrane, referred to as membrane domains or corrals. Since SPT relies on sparse labeling and captures only a minute fraction of receptor dimer events, an important aspect of the spatial model presented here is the explicit consideration of the impact of these new measurements on population dynamics. The spatial model also yields new insight into the activation states of individual monomers after ligand addition, as they cycle through rounds of dimerization, asymmetrical kinase activation and phosphorylation/dephosphorylation.

3.3 MATERIALS AND METHODS

3.3.1 Mathematical Modeling

This ErbB1 model was used as the flagship example while developing the Spatial Stochastic Model, as described in Chapter 2. In this instance, an electron micrograph of immuno-gold labeled ErbB1 receptors was used to define domains and receptor density for the simulation space. This implementation of the SSM includes receptor diffusion, domain escape, and four different reactions. *Diffusion:* At each time step, individual monomers can diffuse or react. If unrestricted by domains, receptors freely diffuse. Receptors also freely enter domains, with defined values for restriction from exit from domain boundaries (Table 3). Escape probabilities are determined through fitting, to arrive at similar receptor cluster distributions at any point during the simulation process; this parameter was validated by the Hopkins spatial statistic (Hsieh et al, 2008; Zhang et al, 2008) and by comparison of jump size distributions with experimental values from single particle tracking (Low-Nam et al, 2011). *Reactions:* Four possible reactions are possible: dimerization (2^{nd} order), dissociation (1^{st} order), phosphorylation (1^{st} order), or dephosphorylation (1^{st} order). After moving, each monomer's position is scanned for other receptors within the binding radius; dimerization likelihood within this radius is based upon a modification of the Smoluchowski approach (Andrews & Bray, 2004) and calculated from dimer estimates in Martin-Fernandez et al. (2002). Dimer dissociation is implemented through a probability calculated using the dimer off rate and the simulation time step. When dimers dissociate, the monomers are assigned an unbinding radius to minimize the occurrence of an unrealistic amount of repeated interactions. These binding and unbinding radii take into account the kinetics and diffusion of the receptors. Using

Species	Diffusion Coeff [$\mu m^2/s$][a]	Dimer On Rate [mm^3/s] [*1E-04][c]	Dimer Off Rate [1/s][a]	Domain Exit Rate [1/frames][a]	Phos Rate [1/s][b]	Dephos Rate [1/s][b]
LR	0.0512	---	---	0.0121	---	---
LRP	0.0512	---	---	0.0121	---	0.13
R	0.0512	---	---	0.0183	---	---
RP	0.0512	---	---	0.0183	---	0.13
LRLR	0.0191	0.9138	0.273	0.00874	0.0733	---
LRR	0.0191	0.9138	0.738	0.00874	0.0733	---
RR	0.0191	0.9138	1.24	0.00874	0.0733	---
LRPLR	0.0191	0.9138	0.273	0.00874	0.0733	0.13
LRPR	0.0191	0.9138	0.738	0.00874	0.0733	0.13
RPR	0.0191	0.9138	1.24	0.00874	0.0733	0.13
LRPLRP	0.00563	0.9138	0.273	0.00874	0.0733	0.13
LRPRP	0.00563	0.9138	0.738	0.00874	0.0733	0.13
RPRP	0.00563	0.9138	1.24	0.00874	0.0733	0.13

Table 3: ErbB1 Model parameters. Set of experimental parameters used for each of the species during simulations. a – Low-Nam et al (Low-Nam et al, 2011), b – Kleiman et al (Kleiman et al, 2011), c – Back-calculated using SMOLDYN (Andrews & Bray, 2004)

the binding and unbinding radius as the inner and outer bound of this region, respectively, minimizes instantaneous rebinding (Andrews & Bray, 2004). Phosphorylation occurs only during dimerization intervals and is based upon a rule where one monomer in the dimer is assigned at random as the activator and the other monomer in the dimer is the receiver. Phosphorylation is estimated as a 1st order reaction based on the assumption of excess phosphate. Dephosphorylation occurs at the same rate irrespective of monomer or dimer state (Table 3). Supporting equations and mathematical methodologies are available in the Supporting Material.

3.3.2 Single Particle Tracking

Detailed methods for tracking and analyzing erbB1 motion are described in Low-Nam et al. (Low-Nam et al, 2011). In brief, erbB1 were tracked with two-color quantum dots (585 and 685 QDs, Molecular Probes) conjugated with either VHH monomeric antibody fragments (non-competing with ligand) or with EGF-conjugated QDs. A431

breast cancer cells were serum starved for a minimum of 2 hrs and observed on an Olympus IX71 inverted microscope equipped with a 60 x 1.3 N.A. water objective and an electron multiplying CCD camera (Andor iXon 887). Samples were maintained at 34-36^{o}C by an objective heater (Bioscience Tools, San Diego CA). QD probes were applied at picomolar concentrations to achieve sparse labeling required for single particle tracking. A 3-state Hidden Markov model was used to identify transition rates between 2 distant monomers (free), co-confined pairs and dimerized receptors. From these rates, the states of receptor pairs in the raw data could be extracted.

3.4 RESULTS AND DISCUSSION

In this work, we began with modifications to our existing spatial stochastic model (Hsieh et al, 2010). Receptors are represented as discrete particles and move through a 2-D simulation space with Brownian motion and under periodic boundary conditions. An improvement in the model is the use of modified Smoluchowski dynamics to govern reactions, as described in Methods. Similarly to Hsieh et al (Hsieh et al, 2010; Hsieh et al, 2008), this simulation approach follows the molecular transformations and Brownian motion of individual particles; however, each dimerization and dissociation reaction type is implemented using a single, pre-calculated geometric parameter, yielding both faster execution and increased physical accuracy.

Dimerization reactions in this simulation environment are diffusion-limited. Individual particles move independently and randomly at each time step, with normally distributed jump sizes. At the end of each move, a scan of the surrounding area within a defined radius of the particle determines where a binding event will occur. This “binding radius” was based upon simulations that reproduce results of Martin-Fernandez et al

(Martin-Fernandez et al, 2002) and takes into account measured reaction rates and diffusion coefficients (Low-Nam et al, 2011), with 1 μsec simulation time steps. The model tracks all particles in the simulation at every step. Rather than monitor binding of ligand, these simulations are initiated with a predetermined percent of ligated receptors as a simplification strategy. Table 3 summarizes the experimental values for dimer off-rates, phosphorylation and dephosphorylation, used to calculate probabilities for events to occur at each time step in the simulation.

The conformational states of erbB1 are specifically represented by dimerization rules in the model, as illustrated schematically in Figure 3.1A. In the absence of bound ligand, receptors (R) are presumed to predominantly assume the bent state, with a 1% probability at each time step of fluxing to the open state that exposes the dimerization arm (Hsieh et al, 2008; Schlessinger, 2002). Ligand-bound receptors (LR) are assumed to be in the extended conformation as long as ligand is bound. Thus, there are three possible types of erbB1 homodimers: two ligand-bound receptors (LRLR), one ligand bound receptor and one non-liganded receptor (LRR), and two non-liganded receptors (RR, the "preformed" dimer state).

An important feature of the model is the introduction of membrane domains that transiently confine receptors. Figure 3.1B-C illustrates how the area and distribution of domains are initialized based upon immunogold-labeling of erbB1 decorating the membrane of A431 breast cancer cells. As described in Methods, both domain location and receptor density are imported directly from EM images through a graphical user interface. By limiting the probability of exit from domains, receptors are confined within them for discrete periods but explore much of the membrane landscape over a period of

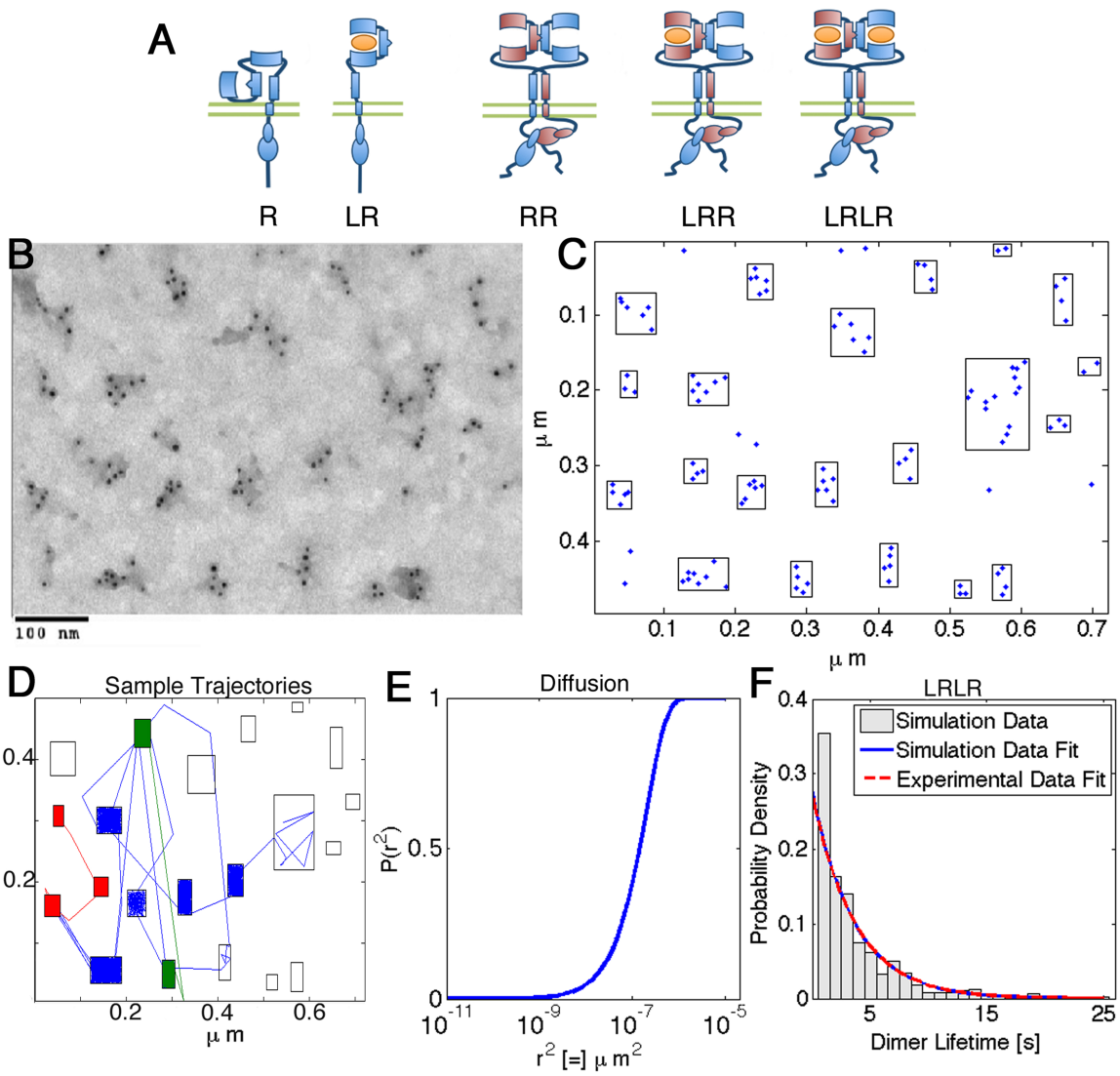

Figure 3.1: ErbB1 species, simulation space, diffusion and off-rate validation (A) Monomer and dimer species accounted for in the spatial stochastic model. R is a resting monomer with no ligand bound. Resting receptors spend 99% of simulation time in a tethered conformation, with 1% probability to flux to the extended conformation. LR is a ligand bound monomer and stabilized in the open conformation. RR is a preformed dimer, formed by encounters between two monomers in the open conformation. LRR is comprised of one unliganded monomer and one liganded monomer. The LRLR dimer is comprised of two ligand bound monomers. (B) TEM image used to initialize the starting positions of ErbB1 receptors and estimate size and density of confinement zones. (C) Simulation interpretation of the TEM image, including static confinement zones in black boxes. (D) Sample trajectory of three different receptors over a 4-minute simulation. (E) Monomer diffusion coefficient calculated from simulation data. Simulation diffusion coefficients match the diffusion coefficients from SPT experiments. (F) Histograms of dimer lifetimes for 2:2 dimers. Each histogram is fit to determine the specific dimer off rate. The red line is the simulation data fit and the blue line is the experimental data fit (Low-Nam et al, 2011).

seconds to minutes (Figure 3.1D). Diffusion coefficients used in the model are based upon SPT measurements (Low-Nam et al, 2011). ErbB1 monomers are assigned the fast diffusion rate of 0.0512 $\mu m^2/s$ for unconfined receptors, slowing to 0.0191 $\mu m^2/s$ upon dimerization. Fully phosphorylated dimers further slow to 0.00563 $\mu m^2/s$, approximating the slowdown attributed to assembly of docking partners and remodeling of the local environment (Low-Nam et al, 2011). The diffusion rates for unphosphorylated dimers were based upon tracking of erbB1 dimers in the presence of the kinase inhibitor PDI53035 (Low-Nam et al). It is noteworthy that we did not assign the slowest diffusion rate to partially phosphorylated dimers that could also slow further when recruiting docking partners, based on comparisons indicating that implementation of further slowdown had no significant impact on the results.

Figure 3.1E summarizes the spread of jump sizes for receptors diffusing and dimerizing within the domain-studded simulation landscape, reported as a CPA (cumulative probability analysis) plot. This analysis compares favorably with CPA plots generated from single particle tracking data for erbB1 bound to QD-probes (Low-Nam et al, 2011). Figure 3.1F shows that the distribution of lifetimes for simulated dimers also closely matches experimental data. The model thus captures the essential features of anomalous diffusion, as well as the stochastic nature of dimer dissociation, observed for erbB1 receptors in living membranes.

3.4.1 Membrane domains promote repeated interactions between monomer pairs

Figure 3.2A illustrates the reproducible observation that pairs of erbB1 monomers, tracked with 2 colors of QD-EGF, can bind and rebind multiple times during live cell

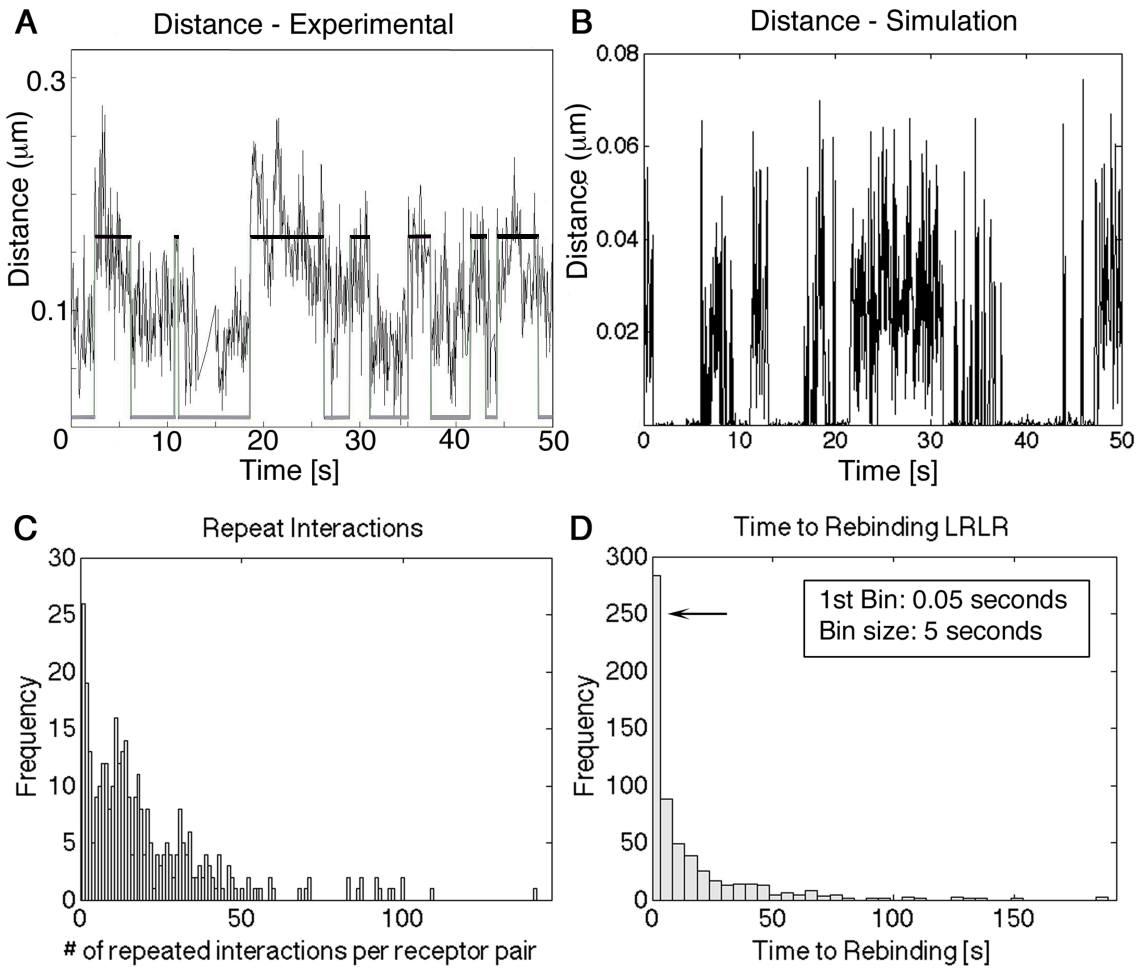

Figure 3.2: Membrane domains influence repeat interactions between receptors. (A) Separation distance over time between two QD-labeled receptors during a single particle tracking experiment. The receptors are initially in a dimer state, dissociate and redimerize several times, as indicated by the state line overlay. (B) Separation distance over time between two ligand bound receptors during a simulation demonstrating the same pattern of repeat interactions. (C) Summary of repeated interactions over an entire simulation for all possible receptor pairs. A few individual receptors interact with one another more than 100 times during a single 4-minute simulation. (D) Time between rebinding interactions of two receptors are shown for LRLR dimers. Many rebinding interactions occur below the frame rate, 20 frames/second, used in SPT experiments (Low-Nam et al, 2011) as indicated by the arrow. While most rebinding incidents occur within 50 seconds, time to rebinding can occur more than 150 seconds later.

imaging. This characteristic behavior has been attributed to co-confinement, based upon the unlikely probability of repeat encounters if dissociated monomers diffuse rapidly away from their original contact site (Low-Nam et al). We tested this notion by examining the trajectories and binding events between receptors in the spatial stochastic model, using a simulation space with membrane domains and 50% ligand-bound receptors. Representative results are shown in Figure 3.2B, where two receptors interact multiple times during a 50 second simulation.

Figure 3.2C reports results of this analysis applied to the entire population of receptors in the simulation space over a 4 minute time course. The number of repeat interactions between each pair of receptors varies broadly, with a high value of 141 binding interactions between a given pair.

Another prediction arising from these simulations is the average time to rebinding. A large number of rebinding reactions (28%) occur within 0.05 seconds (Figure 3.2D, arrow), which is equivalent to the frame rate of the data collection in (Low-Nam et al). Since simulation results are analyzed with millisecond resolution, it suggests that the number of repeated encounters may be underrepresented during image acquisition.

3.4.2 Implications of the asymmetric model for receptor transphosphorylation

We next consider the implications of asymmetric kinase orientation within erbB1 dimers. The cartoon in Figure 3.3A illustrates the basic scheme used to create rules for trans-phosphorylation when kinase activation is restricted to only one monomer in a given pair. Here, the N-lobe of the "activator" monomer is in contact with the C-lobe of the "receiver" monomer. We make the theoretical assumption that the now active

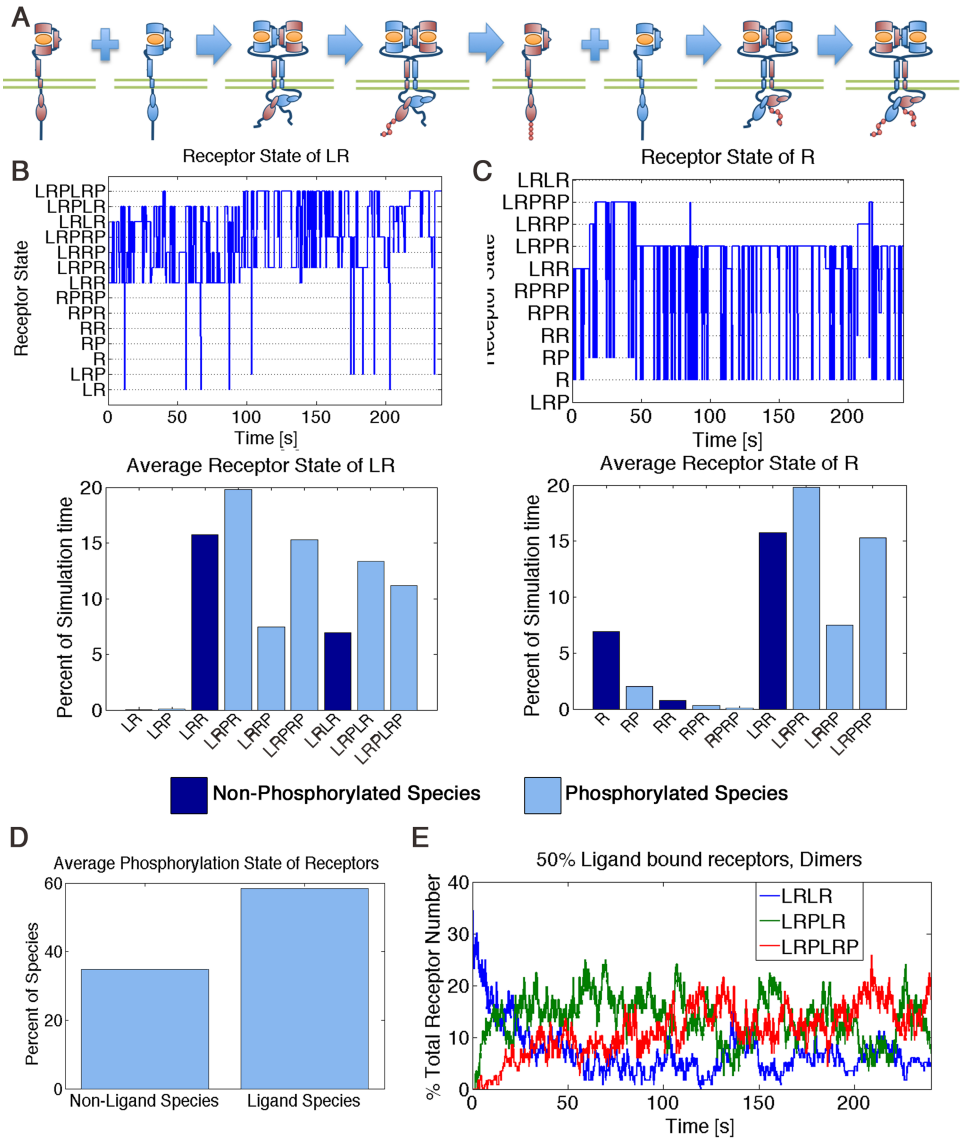

Figure 3.3: Impact of asymmetric receptor phosphorylation (A) A model for receptor phosphorylation shuffle. When a dimer forms, due to the configuration of the N and C lobes, only one tail of the dimer may be phosphorylated. In order for a dually phosphorylated dimer to occur, a phosphorylated receptor and an unphosphorylated receptor must dimerize in the correct configuration such that the unphosphorylated receptor is phosphorylated. (B) Example of a ligand bound receptor state over a 4 minute simulation (top) and the average percent of time spent in each state for all of the ligand bound receptors during the simulation (second row). Ligand bound receptors spend the majority of time in the dimer state. (C) Sample receptor state of a non-ligand bound receptor (top) and the average percent of time spent in each state for all of the non-ligand bound receptors during the simulation (second row). Non-ligand bound receptors also spend a large fraction of time in the dimer state, however they are also found to be in the monomer state much more often than ligand bound receptors. (D) Phosphorylation state of non-liganded and liganded species, independent of receptor state. Less than 40 percent of non-liganded species are phosphorylated, on average, compared to the almost 60 percent of phosphoylated liganded species. (E) Percent of LRLR dimers in different phosphorylation states. LRLR is an unphosphorylated dimer (blue), LRPLR is a singly phosphorylated dimer (green), and LRPLRP is a dually phosphorylated dimer (red). A quasi-steady state is reached in the first minute of the simulation.

"receiver" then trans-phosphorylates its partner; the probability of this enzymatic modification is a function of the dimer lifetime for the pair.

This fundamental premise leads to an interesting prediction: As dimers dissociate and rebind in a stochastic process, it improves the likelihood that each erbB1 monomer has the opportunity to be both receiver and activator. The predicted outcome of this receptor "shuffle" process is illustrated in Figure 3.3B, in the context of a simulation with 50% of receptors bound to ligand at the onset. The graph traces the transition states of a single ligand-bound erbB1 receptor in the simulation space over 250 seconds. Collectively, the ligand-bound receptors in this simulation achieved the dimer state approximately 90% of the time. The predominant dimer type is LRR, due to an equilibrium shift from equal amounts of available LR and R monomers to an equilibrium that favors R monomers (See Figure A.2). Receptors cycle rapidly through all possible dimerization and phosphorylation states, spending 58% of the time as a phosphorylated species (Figure 3.3D).

In contrast, Figure 3.3C tracks the transition states of an unliganded receptor in the same simulation. The unliganded receptors in this simulation participated in dimer events frequently, spending only 9% of the simulation period as free monomers. However, due to the short dimer lifetimes for RR and LRR, only 35% of unliganded receptors are phosphorylated on average (Figure 3.3D).

In order to reconcile transient interactions with sustained signaling, we next analyzed the potential for accumulation of phosphorylated dimers over the same stochastic simulation time course (Figure 3.3E). At early times, the asymmetric model predicts that the predominant dimer state is LRPLR, where only one erbB1 monomer is

phosphorylated. Dimers achieving phosphorylation of both liganded monomers (LRPLRP) reach similar levels with a short delay (red traces in Figure 3.3E). We conclude that rapid receptor re-encounters permit the system to quickly reach equilibrium, providing a significant pool of phosphorylated receptors for recruitment of signaling partners.

In Figure 3.4, we compare steady state phosphorylation for liganded (LRP) and unliganded (LP) receptors over a range of ligand doses and for two different receptor densities. Results for the high density situation are shown in Figure 3.4A, again for A431 cells where the erbB1 gene is amplified and there are an estimated 4 million receptors per cell. At low ligand doses (10-20% occupancy), between 30-40% of the phosphorylated species are unliganded receptors (RP) that interacted with liganded receptors (LRP). As the ligand dose increases, the ratio drops dramatically without raising the overall levels of phosphorylation. The failure to achieve 100% phosphorylation is due to the combined effects of phosphatase activity and the lower availability of free monomers. Plots in Figure 3.4B report ratios of phosphorylated species where erbB1 expression levels were more normal at 30,000 receptors per cell. In this case, the simulation landscape was initialized with erbB1 receptor distributions acquired from immuno-gold labeled Hec50 cells (see Figure 3.5C). At the lowest doses of ligand (10-30%) occupancy, almost 50% of the phosphorylated species are unliganded receptors. We attribute this to the lower availability of liganded monomers in the sparsely populated membrane. These results offer insight into the lateral propagation hypothesis of Bastiaens and colleagues (Verveer et al, 2000) and the observations that 1:2 dimers are signaling competent (Liu et al, 2012). They suggest that initiation of a global response by low doses of ligand is unlikely

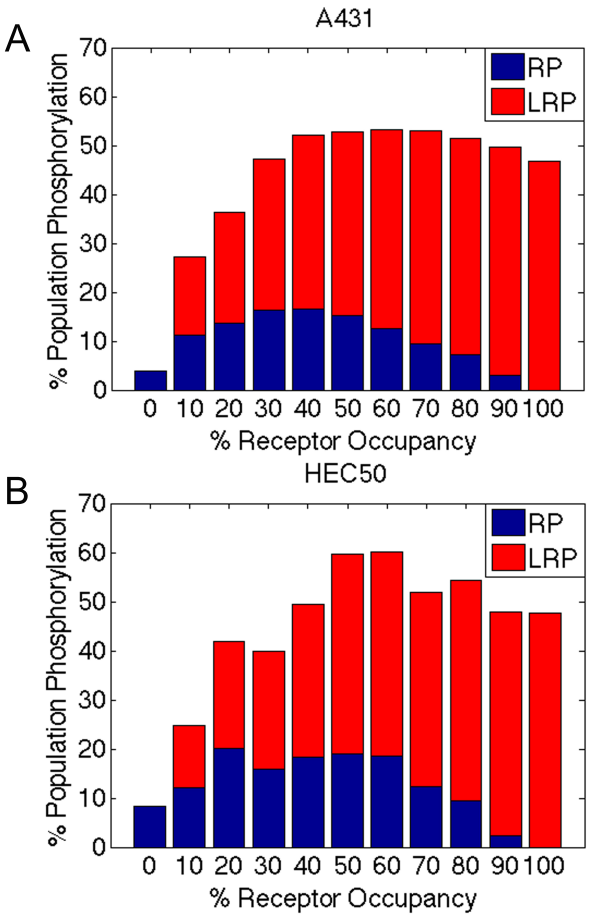

Figure 3.4: Ratio of Phosphorylated receptors, LRP to RP. (A) Percent of phosphorylated LR and R for increasing amounts of receptor ligand occupancy for A431 cells. Initially, at low levels of ligand bound receptor, the percentage of phosphorylated non-ligand bound receptors increases. As liganded receptor percentage increase, unliganded receptor phosphorylation decreases. (B) Percent of phosphorylated LR and R for increasing amounts of receptor ligand occupancy for HEC50 cells. A similar trend of LRP and RP are seen for HEC50, however the sparseness of the receptors on the membrane creates a larger deviation between simulations.

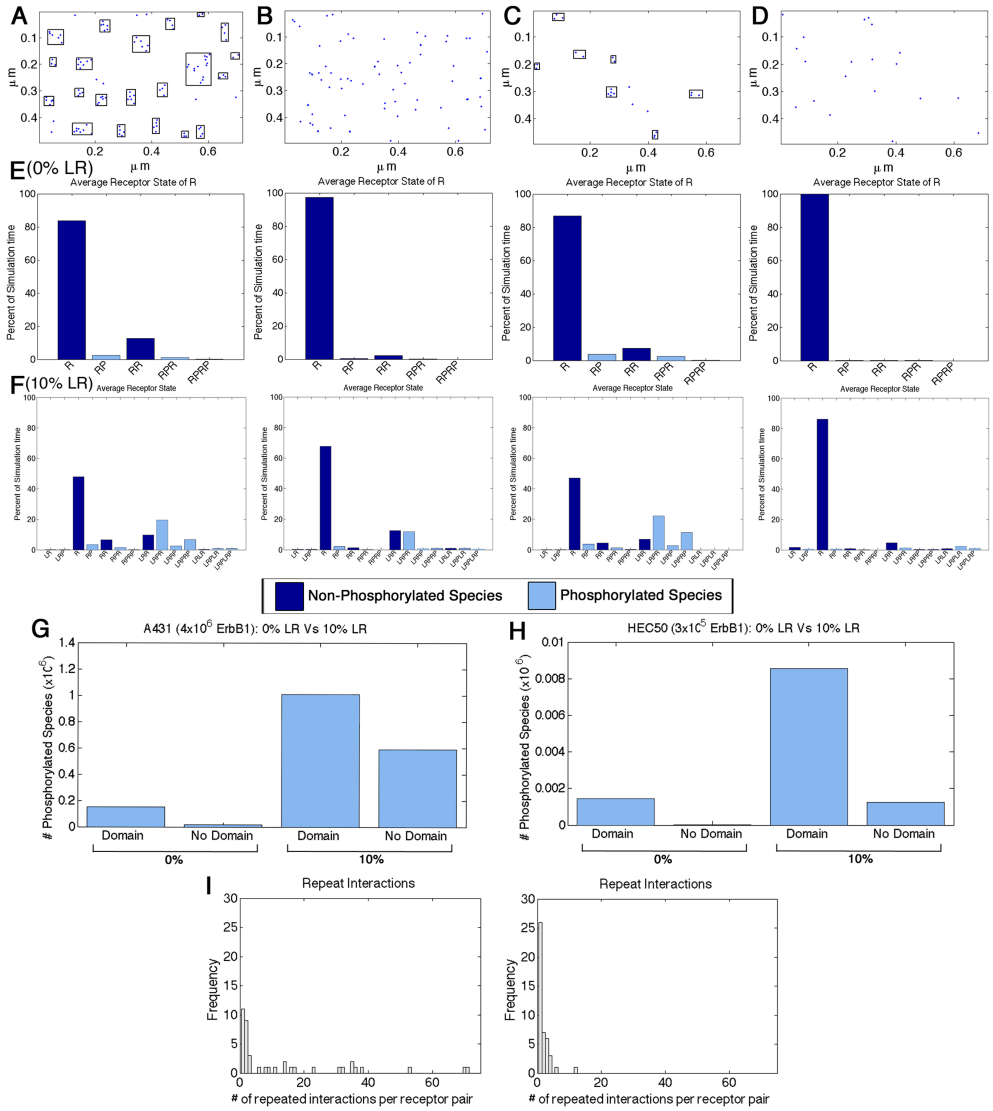

Figure 3.5: Membrane landscape impacts receptor state. (A)-(D) Receptor density and distribution for simulations. (A) Initial simulation space imported from an immuno-gold labeled EM image of an A431 cell. Static confinement zones based on receptor cluster size are included. (B) Randomized distribution of the same number of receptors in (A), after diffusion simulations in the absence of confinement zones. (C) Initial simulation space imported from an immuno-gold labeled EM image of an HEC50 cell. Static confinement zones based on receptor cluster size are included. (D) Randomized distribution of the same number of receptors in (C), after diffusion simulations in the absence of confinement zones. (E) Receptor state for simulation conditions represented in the matching simulation space and 0% ligand bound receptors. The presence of domains impacts how often receptors will encounter one another. The simulations with domains present have a higher rate of dimer occurrence, as well as a large number of phosphorylated receptor species. (F) Receptor state for simulation conditions of 10% ligand bound receptors and corresponding simulation space. Similarly to the 0% ligand bound simulations, the occurrence of dimers and phosphorylated species is increased in the presence of domains. The presence of ligand also allows for the formation of species not present in 0% ligand bound receptor simulations. (G)-(H) Number of phosphorylated species, on average during a simulation, present and scaled to whole cell values for A431 cells (G) and HEC50 cells (H). The presence of domains has a clear impact on the number of phosphorylated species. (I) Repeated interactions of receptors on HEC50 cells with 10% ligand bound receptors present. Simulations with domains and without domains were performed.

when the fast dephosphorylation rates measured by Kleiman et al. (2011) are coupled with fast off-rates for LRR (Low-Nam et al, 2011).

3.4.3 The membrane landscape impacts receptor state

Our next goal was to evaluate the impact of membrane domains and receptor density upon phosphorylation efficiency, integrating both the improved dimer life-time measurements and the asymmetric model. Results are shown in Figure 3.5, where the panels in A-D illustrate the four different conditions initialized into the simulation landscape. We first compared the impact of domains upon the rate of so-called "pre-formed" dimers that occur in the absence of ligand. These events rely on encounters between monomers that have both randomly fluxed up to the extended conformation; since each monomer is assumed to flux at a rate of 1%, there is a 0.01% probability for dimerization at each encounter. Results are compared for ligand-less erbB1 diffusing and transiently getting trapped in domains, versus the same number of receptors diffusing with unrestricted Brownian motion. Results show that domains are particularly influential on the predicted levels of "pre-formed" dimers. As described above, the binding radius was parameterized based upon the observations of Martin-Fernando et al. (2002), who estimated steady state levels of pre-formed dimers on A431 cells at 14%. Thus, as expected, simulations are consistent with this value in the corresponding domain landscape, Figure 3.5E. When domains are removed, dimerization reaches levels of only 2%. At normal receptor density represented by Hec50 cells, up to 10% percent of receptors achieve dimerization in the domain-studded landscape. Remarkably, dimerization of unliganded receptors at this lower density is a very rare event in the absence of domains, with estimates below 0.2%.

Figure 3.5F next compares the dimerization frequencies where 10% of the receptors are ligand bound and subject to the same four initial conditions (high density +/- domains; normal density +/- domains). Results again demonstrate the potential influence of domains, which is particularly impactful on the rate of dimer formation at normal receptor expression levels.

Results in Figure 3.5G-H illustrate the relative impact of domains and receptor density on signaling output, represented by the number of receptors predicted to be phosphorylated at steady state. In the case of high receptor density (Figure 3.5G), up to 3% (~150,000) of receptors are phosphorylated in the absence of ligand on A431 membranes with domains. At 10% ligand occupancy, this value rises dramatically to an estimated 1 million phosphorylated receptors. In the absence of domains, these estimates drop to 18,000 and 587,000 phosphorylated receptors respectively.

For the case of Hec50 cells with normal erbB1 receptor density (Figure 3.5H), predicted values of phosphorylation attributed to pre-formed dimers is modest even in the presence of domains, at only 1400 phosphorylated receptors. Without domains, receptor phosphorylation of ligand-less receptors is exquisitely low (17 total). Values in the case of 10% ligand occupancy are also reported in Figure 3.5H, with 8,500 phosphorylated receptors in the domain landscape and only 1,200 in the absence of domains.

As a final demonstration of the impact of domains, Figure 3.5I compares the predicted frequencies of repeated interactions in Hec50 membranes, with and without domains and 10% ligand occupancy. Repeated interactions occur often when receptors are co-confined, even at this low density of receptors. In the absence of domains, repeated interactions between the same pair of receptors are much more rare events.

3.5 CONCLUDING REMARKS

Dimerization is a key event for many growth factor receptors, including erbB1 and its closely related family members (Lemmon & Schlessinger, 2010). Previous work by us and others have established that erbB1 dimerization is rapidly reversible (Chung et al, 2010; Kawashima et al, 2010; Low-Nam et al, 2011), leaving open important questions regarding the sustainability of signaling. Here we specifically consider receptor dimerization as a diffusion-limited process, with an emphasis on the impact of receptor co-confinement in plasma membrane domains or "rafts". Our approach is based upon mathematical modeling, using a spatial stochastic framework that incorporates the concepts of membrane domains. This approach was validated by its close approximation of receptor diffusion characteristics, including the range of jump distributions measured by single particle tracking. In addition to experimentally determined diffusion behavior, model parameters for dimer dissociation and phosphorylation/dephosphorylation are estimated from quantitative measurements in live cells (Hsieh et al, 2010; Kleiman et al, 2011; Low-Nam et al, 2011). As suggested in our earlier work (Hsieh et al, 2008; Radhakrishnan, 2010), simulations confirm that transient receptor domain confinement can effectively raise local receptor density and enhance the likelihood for productive receptor encounters.

This work has strong implications for the field of membrane biology, where the influence of "receptor clustering" remains a matter of considerable debate. ErbB1 and its family members are among the best studied examples of plasma membrane nanoclustering, with evidence for erbB1 homoclustering in resting cells from a wide variety of techniques including electron microscopy (Hsieh et al, 2010; Hsieh et al, 2008;

Yang et al, 2007), scanning near-field optical microscopy (Nagy et al, 1999), homo-FRET (Yeow & Clayton, 2007), cross-correlation (Costantino et al, 2005; Keating et al, 2008), proximity ligation assay (Soderberg et al, 2006), multispectral plasmon coupling microscopy (Wang et al, 2011), number and brightness (Nagy et al, 2010), and single molecule techniques (Ariotti et al, 2010; Low-Nam et al, 2011; Orr et al, 2005). The phenomenon of membrane protein clustering crosses many cell types. A partial list of examples include MHC molecules (Lavi et al, 2012; Singer & Nicolson, 1972), C-type lectins and viral proteins (Cambi et al, 2004; Itano et al, 2012), TCR, BCR and Fc receptors (Lillemeier et al, 2006; Pierce & Liu, 2010; Wilson et al, 2000), CD36 scavenger receptors (Jaqaman et al, 2011), and GPI-anchored proteins (Brameshuber et al, 2010; Varma & Mayor, 1998).

The observation of nanometer scale proximity of membrane proteins, typically from microscopy methods, is sometimes interpreted as a reflection of oligomerization state. Here, we do not make the assumption that "clusters" observed by immunoelectron microscopy are accurate reporters of the oligomeric state of erbB1. Rather we assume that these images capture a mix of non-random receptor distributions that principally result from monomers diffusing in and out of membrane domains. Productive encounters between monomers can lead to formation of dimers. Since we have yet to experimentally observe or quantify larger erbB1 oligomers (Clayton et al, 2007) with single particle tracking, we do not explicitly consider that interesting possibility here.

There is evidence that membrane domains arise through complex mechanisms, including cytoskeletal barriers (Andrews et al, 2008; Jaqaman et al, 2011; Kusumi et al, 1993; Lavi et al, 2012), the partitioning of saturated lipids and cholesterol (Simons &

Gerl, 2010), and ionic protein-lipid or protein-lipid interactions (Douglass & Vale, 2005; Lillemeier et al, 2006; Spira et al, 2012; van den Bogaart et al, 2011). Due to this complexity, we do not make assumptions here about the primary mechanism underlying the domains that cause erbB1 clustering. The assignment of membrane domain area based upon EM images can be considered a "coarse-graining" approach, where clustering is both maintained throughout the simulation period and satisfies the essential characteristics observed experimentally for receptor motion. In our current simulation framework, domains are held to be static in size and location. This strategy lowers computational costs and follows the observation of Douglass and Vale (2005) that some slow-diffusing membrane proteins can serve as reporters for relatively stable domains. However this simplification likely does not reflect the true dynamic nature of protein-rich domains, that may diffuse as entities in the membrane, themselves encountering cytoskeletal barriers and cycling between growth and dispersion at the nanometer or submicron scale (Lavi et al, 2012).

Our model explicitly considers the mounting experimental evidence for erbB structural rearrangements associated with dimerization. Like the integrins, the extracellular domains of erbB receptors are now well known to exist in both bent and extended confirmations (Burgess et al, 2003; Cho & Leahy, 2002; Ferguson et al, 2003; Ferguson et al, 2000; Garrett et al, 2003; Ogiso et al, 2002), where ligand binding stabilizes the upright form and exposes the dimerization arm. To integrate this concept into mathematical models, we assume that unliganded receptors are predominantly in the bent confirmation and that ligand receptors are fixed in the dimerization competent conformation (Hsieh et al, 2010; Hsieh et al, 2008). A primary goal of the current study

was to incorporate the critical discovery that erbB catalytic activation is dependent upon an asymmetrical orientation of their kinase domains (Jura et al, 2009; Macdonald-Obermann & Pike, 2009; Zhang et al, 2006). These landmark papers established that contact of the N-lobe of the "activator" with the dimer partner's C-lobe relieves autoinhibition of the kinase domain solely in the "receiver" (Zhang et al, 2006). Conclusions of these crystallographic structure studies have supported by electron microscopy analysis of negatively-stained full length EGFR, in the presence and absence of ligand and/or kinase inhibitors (Lu et al, 2012; Mi et al, 2011). These studies indicate that the conformational orientations of dimerized erbB kinase domains are dominated by the active asymmetrical orientation, as opposed to the inactive symmetrically orientation (Landau et al, 2004), although kinase inhibitors can shift the class averages for the two orientations.

We consider the implications of the asymmetric erbB1 activation scheme in its simplest form, by assuming that during the lifetime of the dimer only one member of the dimer pair becomes catalytically competent for transphosphorylation of its partner. Consistent with evidence that dimers composed of 1ligand:2receptors are signaling competent (Liu et al, 2012), the activation state of erbB1 in our simulations is not governed by ligand occupancy *per se* but rather by the lifetime of dimers determined experimentally (Low-Nam et al, 2011). The probability for productive interactions is highest for 2:2 receptors, since the off-rate is slowest, followed by 1:2 receptor pairs and by 0:2 preformed dimers that have very fast off-rates. We do note that the number of phosphorylated receptors is increased above the total value for ligand-bound receptors, through repeated interactions and the productivity of 1:2 dimers. This amplification,

combined with transient dimerization, does allow for phosphorylation of unliganded receptors in 1:2 dimers that then dissociate and later interact with other unliganded monomers. However, the shorter lifetimes and reduced interaction probability associated with unliganded receptors results in very few productive 0:2 dimer events.

Since our simulations are initiated with a fraction of monomers bound to ligand, our current model does not consider the potential for negative cooperativity (Adak et al, 2011; Macdonald-Obermann & Pike, 2009; Tynan et al, 2011). If ligand binding were to be considered in the spatial stochastic model, it would lower the probability for an additional ligand to bind to a 1:2 receptor pair only during its relatively short lifetime (k_{off} = 0.738s^{-1})(Low-Nam et al, 2011).

One notable prediction of the simple asymmetric model considered here is that fast dissociation of dimers effectively promotes signaling, because re-encounters increase the likelihood that each monomer has repeated, equal opportunities to become phosphorylated by the "receiver". We note the recent work of Pike and colleagues, who used a novel luciferase fragment complementation assay to provide compelling evidence for asymmetric and sequential activation of kinases in erbB homo- and heterodimers (Macdonald-Obermann et al, 2012). These authors also raise the possibility that reciprocity could occur during the lifetime of the same dimer event, if the kinase domains can reorient while the monomers remain bound. This intriguing possibility is not explored here, due to lack of information about energetic requirements and feasibility of such a reorientation on the time scale relevant to even the most stable 2:2 dimer (<10 sec).

This work adds to a growing appreciation that cell signaling is markedly influenced by the spatial organization of the plasma membrane, where lateral segregation in the 2D environment influences interactions between signaling proteins and the propagation of positive signaling or associated negative regulatory networks (Casaletto & McClatchey, 2012; Dehmelt & Bastiaens, 2010; Harding & Hancock, 2008). Our simulations predict that ligand-bound erbB1 cycle rapidly through all possible receptor states, generating pulses of signaling competent states. The potential for short-lived components to generate robust, system-level output has been coined "digital signaling" (Harding & Hancock, 2008). We expect that the impact of membrane spatial organization will vary widely in disease and normal settings, even for a single species of receptor such as erbB1. For example, we show here that cells expressing very high levels of erbB1 (typical of gene amplification in certain cancers) are less dependent on domain co-confinement for productive encounters than cells with modest levels of surface receptors. Cell-type variable factors that could alter the stability of domains and extend receptor capture events include lipid composition, the extent and dissociation kinetics of cortical cytoskeletal connections with membrane anchors, and the lipid-protein ratio. Since lipid remodeling, protein macromolecular assembly, and cytoskeletal rearrangements often accompany signaling, the organization of the plasma membrane is subject to alterations over important time and length scales. Highly diffusible products of signaling cascades, such as reactive oxygen species proposed to inhibit phosphatases acting on erbB1 and enhance lateral propagation (Reynolds et al, 2003), would not be subject to the same 2D restrictions. Exploring the impact of the evolving 2D and 3D landscape through creative imaging and mathematical approaches is a future challenge for the field.

3.6 NOTES

This work was published in Biophysical Journal on September 17th, 2013 in Volume 105, Issue 6, Pages 1533-1543.

3.7 ACKNOWLEDGEMENTS

This work was supported by NIH grants R01 GM104973 (to JSE and AMH), R01 CA119232 (to BSW), K25 CA131558 (AMH), NSF CAREER MCB-0845062 (DSL) and P50GM085273 (New Mexico Spatiotemporal Modeling Center). MMP was supported by the NSF INCBN IGERT Fellowship (DGE-0549500). Use of the UNM Cancer Center Microscopy Facility, and NIH support for instruments and staff support, is gratefully acknowledged. The computational platform is accessible by request to the authors.

CHAPTER 4: NON-BROWNIAN FEATURES OF FCεRI DIFFUSION REVEALED BY SPT ARE CONSISTENT WITH A SPARSE STRUCTURE OF ATTRACTIVE MICRODOMAINS

Ádám M. Halász[#,*], Meghan McCabe Pryor[¶,*], Patrick J. Cutler[‡], Bridget S. Wilson[‡], Diane S. Lidke[‡], and Jeremy S. Edwards[†,¶,§,**]

[#]Department of Mathematics, West Virginia University, Morgantown, West Virginia; [¶]Department of Chemical and Nuclear Engineering, University of New Mexico, Albuquerque, New Mexico; [‡]Department of Pathology, University of New Mexico Health Sciences Center, Albuquerque, New Mexico; [§]Department of Molecular Genetics and Microbiology, University of New Mexico Health Sciences Center, Albuquerque, New Mexico; [†]Department of Chemistry and Chemical Biology, University of New Mexico, Albuquerque, New Mexico.

*Equal contribution

**Correspondence: JSEdwards@salud.unm.edu

4.1 SUMMARY

The anomalous diffusion of membrane receptors has been linked to trapping in transient confinement zones, which may be due to restrictions imposed by the cortical cytoskeleton or by lipid- and protein-defined nanodomains. Single particle tracking (SPT) experiments reveal subdiffusion properties, where the apparent diffusion constant decreases with the observation time and reflects confinement at length scales below 500 nm. High resolution imaging of fixed receptors show a non-random distribution, with clusters that range from a few to hundreds of receptors. It is unclear if these clusters are synonymous with lipid rafts or protein islands, which have size estimates in the range of 20-100 nm. Also unknown are the relative exchange rates of receptors diffusing between these clusters. Herein, we investigate SPT data from FcεRI receptors, performed at a rate of 20 frames per second. While the time dependence of the mean square displacement exhibits subdiffusion, the apparent diffusion constant derived from the distribution of displacements over a fixed time duration increases with the jump distance. We show through simulations that this feature can be the consequence of a combination of factors. If the frame interval is comparable to the diffusion time scale through a confining microdomain, free movement inside the domain is not directly observable. We were able to closely reproduce the observed fixed duration distributions from Monte Carlo simulations, performed in a landscape of small, confining domains of ≈100 nm diameter that cover a few percent of the membrane area. These simulations also exhibit subdiffusion. Simulations in a landscape of uniformly semi-permeable barriers but no confining domains produced qualitatively different distributions.

4.2 INTRODUCTION

In the original fluid mosaic model (Singer & Nicolson, 1972), membrane proteins were assumed to move more or less freely along the membrane. As experimental methods improved, it became clear that the plasma membrane has a complex structure with lipid rafts (Edidin, 2001; Simons & Ikonen, 1997; van Meer & Simons, 1982), aggregations of receptors and other membrane proteins (Lillemeier et al, 2006), and elements of the cytoskeleton (Kusumi & Sako, 1996). Studies of the mobility (Axelrod et al, 1976; de Keijzer et al, 2008; Feder et al, 1996; Low-Nam et al, 2011; Schütz et al, 1997) and localization (Bobroff, 1986; Gelles et al, 1988; Ober et al, 2004; Schmidt et al, 1995) of membrane receptors and other transmembrane proteins have been a continued source of information on the rich structure of the cell membrane.

Studies based on single particle tracking (SPT) techniques have revealed deviations from normal Brownian motion (Sako & Kusumi, 1994; Sako & Kusumi, 1995; Simson et al, 1995; Simson et al, 1998), consistent with transient confinement in areas of characteristic length in the range of 300-600 nm. These results give rise to the model of hop-diffusion, where particles are embedded in a network of microdomains separated by actin filaments and other elements of the cortical cytoskeleton; diffusion within a microdomain is normal, but crossing between domains is partially restricted. This model was further refined by theoretical and simulation studies of diffusion in the presence of various types of obstacles, including trapping in corrals, as well as by non-specific binding sites (Niehaus et al, 2008; Saxton, 1995; Saxton, 2008). In general, the predicted behavior is subdiffusion, where the effective diffusion constant decreases as the observation time or distance increase. Complementary to SPT methods that focus on a

small number of molecules at high temporal resolution, protocols such as superresolution and immunogold transmission electron microscopy provide “snap-shot” position data of a large number of molecules of interest. These approaches (Lillemeier et al, 2006; Nagy, 2002; Veatch et al, 2012; Wilson et al, 2000; Wilson et al, 2001; Yang et al, 2007) have revealed clustering of receptors in areas consistent with a structure of quasi-randomly distributed microdomains that in total comprise a small fraction of the cell membrane. Small clusters are observed even in the absence of ligand stimulation, with diameters in the range of 20-100 nm. The connection between the proposed mechanisms underlying subdiffusion and these smaller receptor clusters is not well understood.

Recent results using sophisticated imaging techniques indicate that most yeast membrane proteins tend to segregate, self-organizing into separate domains (Spira et al, 2012). Proposed mechanisms for membrane protein clustering in higher eukaryotes relate it to lipid rafts and/or a corral structure induced by the cytoskeleton. Cytoskeleton-induced corrals are consistent with larger (≈ 200 nm) domains and can preserve, but not induce, local concentrations of specific proteins. A plausible mechanism for the origination of protein clusters relies on vesicle trafficking of proteins newly delivered to the membrane (Gheber & Edidin, 1999; Lavi et al, 2007; Lavi et al, 2012). On the other hand, lipid rafts (Parton & Hancock, 2004; Rao & Mayor, 2005) have a complex chemical structure and may attract specific membrane proteins. Lipid rafts have been generally associated with “nanodomains” as small as 20 nm, which exhibit high mobility (Brameshuber et al, 2010). The variety of membrane associated proteins and of the observed spatial and temporal scales of clustering suggests that there are several, perhaps overlapping mechanisms at work (Saikh & Edidin, 2006). One such mechanism is the

transient confinement (Simson et al, 1998) of specific proteins by specialized microdomains.

Recent SPT results with FcεRI (Andrews et al, 2008) as well as EGFR (Low-Nam et al, 2011) demonstrate transient co-confinement of pairs of receptors, clearly distinguishable from correlated motion due to dimerization. The concept of transiently confining microdomains has been implemented in computer simulations of the movement of membrane receptors. Simulations of diffusion in a landscape of small confining microdomains (consistent with the locations of clusters derived from static TEM images) successfully recapitulate the observed signal initiation kinetics (Hsieh et al, 2010; Hsieh et al, 2008; Pryor et al, 2013). This prompted us to consider the impact of confining domains on diffusion properties of receptors, and compare it with mobility statistics from SPT experiments.

Here we seek a quantitative understanding of anomalous diffusion through comparative results of simulations incorporating transiently-confining domains with authentic SPT trajectories of quantum-dot (QD)-labeled FcεRI receptors. SPT data were taken at intervals of 50 ms, in the absence of ligand stimulation (Figure 4.1). The time dependence of mean square displacements is consistent with subdiffuson but the distribution of displacements at fixed duration exhibits limited movement at short distances and faster diffusion at longer distances, in apparent contradiction with the confinement hypothesis. We show through numerical simulations that this is qualitatively consistent with a structure of small, transiently confining domains in the range of 30-250 nm (comparable to the diffusion length corresponding to frame interval τ). The observed distributions can be closely duplicated by model simulations.

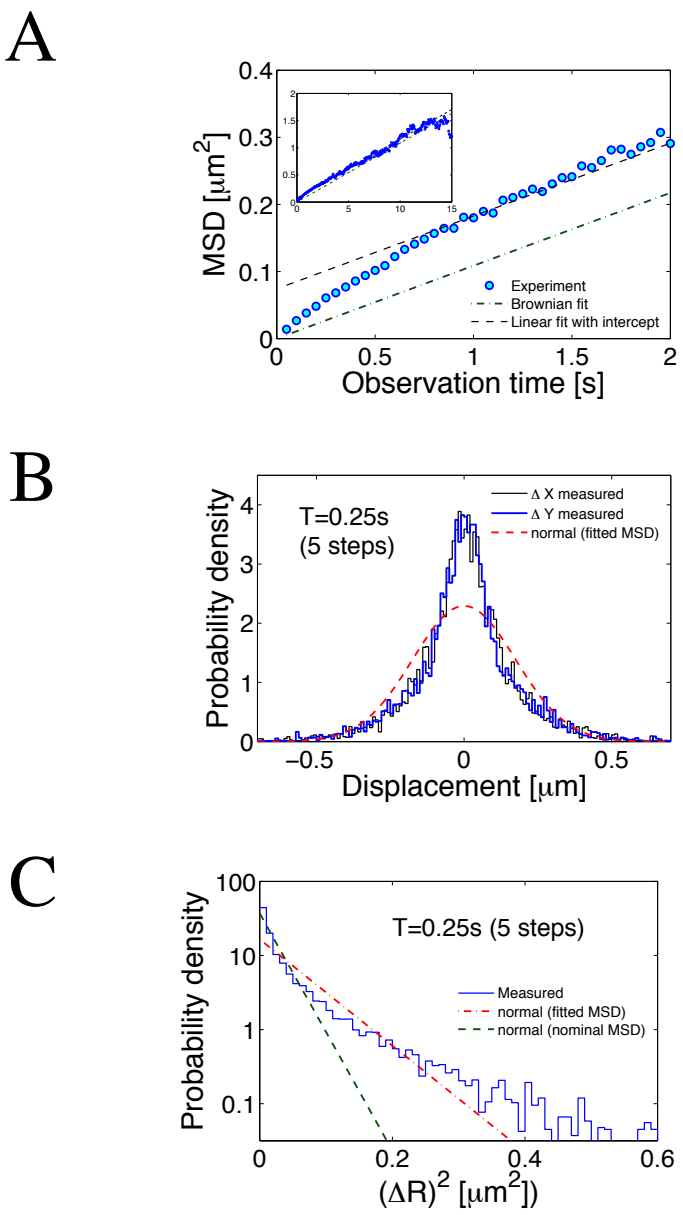

Figure 4.1: Experimental results. Displacement statistics for the aggregate of 1685 reconstructed trajectories, derived from 21 movies. Both the mean square displacement (MSD) as well as the step (jump) distributions over fixed time (duration) exhibit deviations from classic Brownian motion. A. MSD versus the observation time. The dependence is sublinear (has a decreasing slope) for short time intervals. The slope becomes constant for $T_{obs} = 1$ s, up to 14 s; this regime is well approximated by a linear dependence with an intercept. B. Distribution of linear (x and y) displacements over time intervals of 0.25 s, (5 recording intervals). Compared to a normal distribution with the same MSD, there is an excess of short (and very long) jumps. C. The distribution of square displacements at fixed time illustrates the deviation from Brownian behavior more clearly. The logarithm of the probability density (Eq.2) as a function of the value of the square displacement should follow a line with slope $-1/(4DT_{obs})$; instead, the square displacement distributions at fixed duration have a characteristic upward curved shape.

4.3 MATERIALS AND METHODS

4.3.1 Background

Molecules in a homogeneous environment typically execute a random walk, a succession of discrete steps interspersed with random changes of direction and speed. A point particle is said to execute Brownian motion in two dimensions (d = 2) if its displacements (x, y) after time, t, are random and normally distributed, according to the probability density function (PDF)

$$f(x,y;t) = \frac{1}{4\pi Dt}\exp\left(-\frac{x^2+y^2}{4Dt}\right), \qquad (1)$$

where D is the (isotropic) diffusion constant. Eq. (1) implies two easily verifiable properties regarding the displacement $r \equiv (x^2 + y^2)^{1/2}$. First, the distribution $P(r^2)$ of square displacements r^2 after a fixed time t is exponential:

$$P(r^2) = \frac{1}{2\sigma_{xy}^2}\exp\left(-\frac{r^2}{2\sigma_{xy}^2}\right), \qquad (2)$$

where the standard deviation σ_{xy} of the displacement in either the x or y direction satisfies $\sigma^2_{xy} = 2Dt$. Second, the mean square displacement (MSD), defined as the expectation of the square of the displacement vector, $\langle r^2\rangle = \langle x^2\rangle + \langle y^2\rangle$, is proportional to the time (duration), t, over which the displacement takes place. The slope is determined by the diffusion constant,

$$\langle r^2 \rangle = 2\sigma_{xy}^2(t) = 4Dt\ . \qquad (3)$$

Single particle tracking experiments provide estimates of the coordinates (x_k, y_k) of individual particles in a sequence of frames taken at some time interval τ. These can be used to construct distributions of step sizes at fixed observation times (durations) T_{obs},

corresponding to integer multiples of the frame interval ($T_{obs} = \tau, 2\tau, \ldots$), as well as to derive the dependence of MSD values on the observation time.

SPT results often reveal anomalous diffusion that deviates from standard Brownian motion. The presence of obstacles, either a network of barriers (Sako & Kusumi, 1994; Saxton, 1995), or a set of trapping locations (Saxton, 2008), typically results in a behavior that is consistent with unimpeded diffusion at time and spatial scales below that of the obstacles, and slower diffusion at larger scales. The classic signature is subdiffusion, a sub-linear dependence (decreasing slope) of the mean square displacement as a function of time.

Turning to distributions of step sizes over time intervals of a given length, obstacles should result in faster movement over short distances, and limited (slower) movement over long distances. This is to be expected if the characteristic size of the barriers B is larger than the diffusion length, $\sigma = 2\ (DT_{obs})^{1/2}$, associated with the observation interval T_{obs} (given by a frame rate). Traditionally, the $\sigma < B$ condition is considered necessary for the observation of confinement, requiring high-speed recording capabilities (Saxton, 2009). With moderate frame rates, the effect of confinement is readily identifiable, albeit more subtle.

When comparing trajectories obtained from single particle tracking with model simulations, one must take into account the sources of error due to the experimental techniques used to obtain the trajectories. Step size distributions derived from imaging are potentially impacted by a localization error (Pezzarossa et al, 2011; Saxton, 2009), due to the uncertainty affecting the position measurements. The position analysis of quantum-dot (QD) tracking data relies on fitting a curve to the intensity distribution

recorded in several pixels, over the observation time. Position uncertainty adds a normally distributed error to the true position. If this error is independent from the true position, it will result in a normally distributed apparent displacement added to the true distributions, and a constant added to the corresponding MSD. Another source of uncertainty affecting positions is due to time averaging; the signal generated by the light detected by each pixel is recorded continuously, and the intensities reported for each frame are derived from the time-integrated signal over the entire frame interval. In our analysis, we simulated the effect of localization uncertainty and time averaging on the positions of particles (Figure A.4).

Fluorescent imaging techniques rely on the performance of the fluorescent tag (fluorophore) used in the experiments. All fluorophores have a finite lifetime due to photobleaching. Quantum dots provide superior performance in this respect, but are subject to blinking, short, random intervals when they do not emit their characteristic light; thus individual QDs cannot be detected in a subset of the images captured during an experiment. The resulting tracks are fragmented, but can be reliably re-assembled if the density of fluorophores is appropriately low (Smith et al, 2010). The fragmentation results in an incomplete record of the positions of the respective particle; this is taken into account when sampling the displacements at various numbers of multiple steps (Figure A.3C).

4.3.2 Single Particle Tracking (SPT) Data and Analysis

4.3.2.1 Cell preparation, data acquisition, and tracking.

SPT experiments were performed on RBL-2H3 cells labeled with QD-IgE as previously described (Andrews et al, 2008). In brief, QD-IgE binds to its high affinity

receptor (FcεRI) expressed on the mast surface, permitting single molecule tracking. QD-IgE is prepared by mixing biotinylated IgE with streptavidin QDs (Invitrogen) at a 1:1 ratio. Cells were primed with 100 pM of each color of QD-IgE (585 and 655 nm) for 10 minutes at 37 °C. QD-IgE primed cells were imaged using an Olympus IX71 with 436 nm excitation (mercury lamp with 436/10 nm band-pass) and emission collected through a QuadView image splitter (Optical Insights) with 655/40 and 585/20 nm band-pass filters for simultaneous imaging of 2 channels (Low-Nam et al, 2011). Imaging temperature was maintained at 34-36 °C by an objective heater (Bioscience Tools). Data was acquired at 20 fps (50 ms exposure time) for a total of 1000 frames. Single-molecule localization and trajectory connections were performed as previously described (Andrews et al, 2008; Low-Nam et al, 2011). Short tracks were elongated (concatenated after particle identification) using the procedure described in (Low-Nam et al, 2011).

4.3.2.2 Step size distributions.

For each trajectory identified as described above, the X and Y coordinates were recorded for every frame (time step) when an identification of the respective particle was made. We collected x and y displacements for every integer multiple $T_{obs} = N_T \cdot \tau$ of the frame interval τ = 50 ms from N_T = 1 to at least N_T = 200 by selecting pairs of observations of the same particle separated by the respective number of time steps. To ensure statistical independence, the pairs were selected so that the corresponding time intervals for the same N_T would never overlap, but could share endpoints, as illustrated in Figure A.3C.

4.3.3 Numerical Simulations

4.3.3.1 Brownian motion in the presence of barriers

Basic Brownian motion is simulated in a fixed time step algorithm that replicates the mathematical definition. Given the position (X_k, Y_k) of a particle after k iterations, the displacements (x, y) are each selected from the distribution defined by Eq. (1) with t set equal to t_{sim}, the simulation time step. The position vector is then set to: (X_{k+1} , Y_{k+1}) = (X_k + x, Y_k + y). In the absence of obstacles or boundaries, this algorithm is exact.

In order to separate the observation and simulation time scales, and to avoid discretization artifacts in the diffusion process, the *simulation* time step t_{sim} was set as a fraction of the lesser of the observation time T_{obs} and the characteristic time $t_{diff} = B^2/D$ for diffusion through the barrier spacing length unit B:

$$t_{sim} = \frac{\tau_{\min}}{N_{steps}}; \tau_{\min} \equiv \min\left(\frac{2B^2}{D}, T_{obs}\right), \qquad (4)$$

with $N_{steps} \geq 100$. The step distributions had a weak dependence on the simulation time step, but stabilized when $t_{sim} << \tau_{min}$. We verified that our results were essentially unchanged upon decreasing the simulation time step by up to three orders of magnitude (Figure A.4).

Similarly to (Niehaus et al, 2008; Saxton, 1995; Wieser et al, 2007) and others, semi-permeable barriers are implemented as line segments with no thickness. Barriers are characterized by a dimensionless permeability or crossing probability, $p_{cross} \leq 1$. When, during a simulation update, the next position of a particle would result in crossing a barrier, the move is accepted with probability p_{cross} and is rejected otherwise (Figure A.4A). If the move is rejected, the position of the respective particle stays the same. In a departure from similar work in the literature (Heinemann et al, 2013; Saxton, 2007;

Wieser et al, 2007), we let the same barrier have different permeabilities for crossing in one or the other direction. Barriers with 100% permeability are equivalent to free movement. We use asymmetric barriers to model confining domains; crossing into the domain is allowed with permeability 1, but exit from the domain has $p_{cross} \approx 1 - 5$ %.

4.3.3.2 Corrals and confining domains

In all the simulations presented here, the barriers form a rectangular network, with sets of barriers parallel to each axis, as illustrated in Figure A.4B. The network is defined by two sequences of barrier coordinates, $\{X_1^B, \ldots, X^B_{NB}\}$ and $\{Y_1^B, \ldots, Y^B_{NB}\}$, each arranged in increasing order. The simulation area is thus partitioned into rectangles $R_{jk} = [X_j^B, X_{j+1}^B] \times [Y_k^B, Y_{k+1}^B]$, which represent our domains. The rectangular geometry allows for an implementation of the Brownian simulation at a moderate computational cost. The simulation area is subject to periodic boundary conditions.

We used the assignment pattern of barrier permeabilities to emulate two types of membrane landscapes. In the corral configuration (Figure 4.2A, inset), boundaries are uniformly semi-permeable and each of the N^2_B domains represents a corral separated from its neighbors by such boundary segments. In the confining domain configuration (Figure 4.2B, inset), only a subset of the rectangles (one in a patch of N^2_B) has semi-permeable boundaries. The boundaries of a confining domain are semi-permeable (p_{cross} = 1−5%) for particles exiting the domain, and are fully permeable (p_{cross} = 100%) for particles entering; all other boundary segments are turned off (set to full permeability, thus not interfering with particle motion).

The barrier coordinates are specified in terms of an intrinsic distance unit B. In simulations with uniform barrier spacing (Figure 4.2AB), the barriers were located at

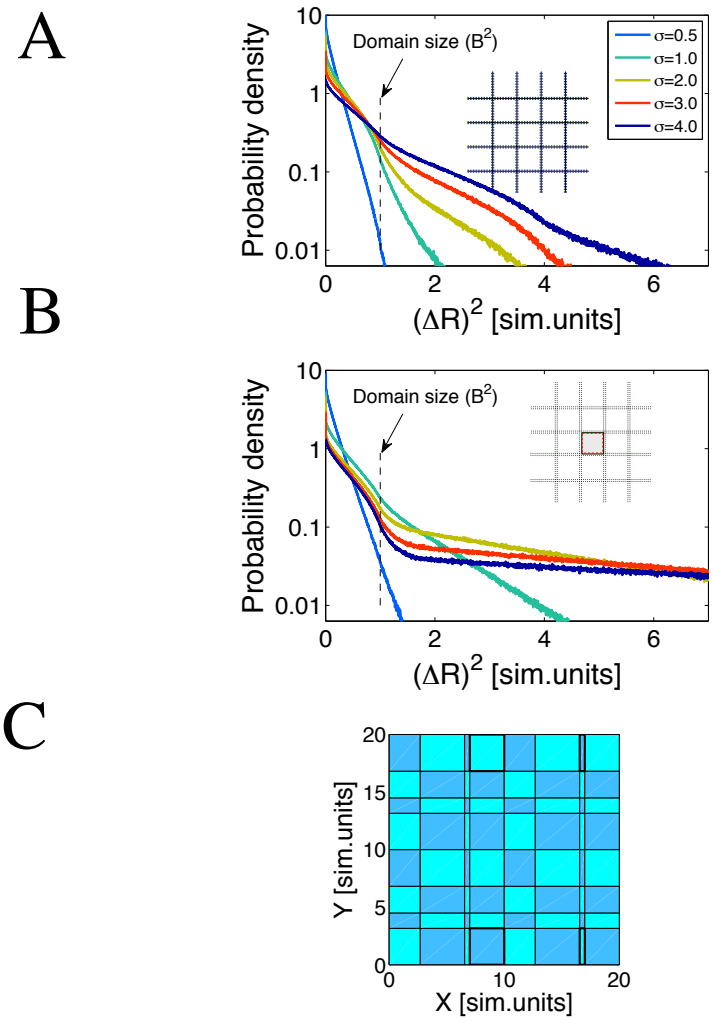

Figure 4.2: Simulations of Brownian motion in the presence of semi-permeable barriers. A. Distribution of individual square displacements (ISD) for different observation times (durations), in a rectangular grid of evenly spaced (B =1 in simulation units), uniformly permeable (p_{cross} = 0.01) barriers. The barriers induce a pattern of alternating faster and slower apparent diffusion. If the diffusion length corresponding to the observation time is smaller than the barrier spacing, the alternation begins with fast diffusion, slowed down at the barrier, then faster, and so on. For larger observation times, the initial fast diffusion regime is washed out , and the curve begins with slow diffusion, followed by fast. B. The clean upward curved shape, not seen with repeated barriers, is typical of the single trapping domain configuration. These simulations also exhibit an initial faster regime. However, this regime is not visible in the simulations shown in Figure 4.4, which did not use identical, square shaped domains. C. Outline of the barrier landscape used in the simulations shown in Figure 4.4. To avoid quasi-periodic behavior, the barrier spacings were chosen from a normal distribution (centered on B = 2.5). The barriers are generally semi-permeable; the domains outlined with thick lines were partially confining (free entrance, small exit permeability). The configuration shown here was repeated using periodic boundary conditions.

multiples of B: $X^B = Y^B = \{0, B, 2B, 3B, \ldots\}$. To avoid spurious effects due to periodicity, we introduced non-uniform barrier spacing selected from a normal distribution centered on the average spacing.

The geometry of the simulations discussed in Figure 4.4 is depicted in Figure 4.2C. The basic patch of 10B ×10B is divided into 4 × 4 = 16 domains (thus $\Delta X^B \approx \Delta Y^B \approx$ 2.5B) , using the following barrier coordinates: $X^B \equiv \{0, 2.70, 6.56, 7.01, 10.00\} \times B$; $Y^B \equiv \{0, 3.16, 4.50, 6.81, 10.00\} \times B$. This primary pattern of barriers was repeated once in each direction, for a total simulation area of 20B × 20B with 64 domains. In each instance of the elementary pattern, one the 16 boxes (initially chosen at random but kept the same throughout the different simulations) was confining. The full 20 × 20 area was subject to periodic boundary conditions.

All our simulations and analysis were implemented in Matlab. In a typical simulation, we used 10^4 particles and a total simulation time to 10^3 times the characteristic time τ_{min} defined in Eq. (4). For corral simulations, particles were initially distributed uniformly in the simulation area. The initial states for confining domain simulations had a proportionally higher fraction of the particles distributed uniformly inside the confining domains.

4.3.3.3 Units, scales, conversions

The simulations are performed using intrinsic (or simulation) time and length units, with a diffusion constant of $D^{(sim)} = 100$ and barrier spacing length scale $B^{(sim)} = 1$. Individual simulation runs are characterized by the configuration type, barrier geometry (X^B, Y^B), permeability p_{cross}, simulation time step t_{sim}, and the observation time T_{obs}. The

dimensionless ratio between the barrier spacing length scale B and the diffusion length corresponding to the observation time,

$$\sigma_{rel} = \frac{2\left(D \cdot T_{obs}\right)^{1/2}}{B} \qquad (5)$$

is independent of the units, hence it also corresponds to the ratio of the physical diffusion length and barrier spacing scale (or domain size parameter), $B^{(phys)}$. In simulation units (where B = 1), σ_{rel} coincides with the diffusion length.

Conversion to physical units requires setting the time and length conversion factors, defined as follows:

$$t^{(phys)} = \alpha_T \cdot t^{(sim)}; x^{(phys)} = \alpha_L \cdot x^{(sim)} \qquad (6)$$

The time conversion factor is determined from the requirement that the observation time in the simulation correspond to the physical one:

$$T_{obs} = \alpha_T T_{obs}^{(sim)} \rightarrow \alpha_T = \frac{T_{obs}}{T_{obs}^{(sim)}} \qquad (7)$$

In principle, the length conversion factor could be determined using a known physical distance, for example by setting the barrier spacing scale B based on the typical linear size of a microdomain. A more practical option is to use the apparent diffusion constant, defined by the slope of the mean square displacement versus time plot,

$$D_{eff}^{(phys)} \approx \frac{1}{4}\left(\frac{d\langle r^2\rangle}{dt}\right)^{(phys)} = \frac{\alpha_L^2}{4\alpha_T} \cdot \left(\frac{d\langle r^2\rangle}{dt}\right)^{(sim)} = \frac{\alpha_L^2}{\alpha_T} \cdot D_{eff}^{(sim)} \rightarrow \alpha_L = \sqrt{\frac{\alpha_T D_{eff}^{(phys)}}{D_{eff}^{(sim)}}} \qquad (8)$$

4.4 RESULTS

4.4.1 SPT tracking confirms FcεRI motion is non-Brownian, with two distinct anomalous features

We analyzed a dataset of 1685 traces reconstructed from high resolution imaging of quantum dots conjugated to IgE (QD-IgE), which attach to FcεRI receptors on unstimulated RBL cells. The data was collected in three experiments that resulted in sets of 686, 319, and 653 trajectories respectively. Each set contains 7 recordings (or movies) of 1000 frames each; each movie resulted in a number of trajectories obtained simultaneously, in the same image field of a biological sample. The images were taken at an interval of τ = 50 ms, corresponding to a frame rate of 20 fps. While the general aspects of the trajectories are those of a random walk (Figure A.3A), the step size distributions (Figure 4.1) deviate from standard Brownian behavior.

We analyzed the displacements of the particles over a range of 1 to 200 consecutive time steps, corresponding to observation times from T_{obs} = 50 ms to 10 s. The mean square displacement (MSD) increases with T_{obs} (Figure 4.1A); the slope initially decreases, then stabilizes at $T_{obs} \geq 1$ s. The expected dependence for Brownian motion (Eq.3) is linear, $\langle r^2 \rangle = 4D \cdot T_{obs}$. A diminishing slope corresponds to an effective diffusion coefficient that decreases with the observation time, from an initial value of D'eff = 0.058 $\mu m^2/s$ to an average of D'eff = 0.033 $\mu m^2/s$. This non-Brownian behavior is generally associated with reduced mobility or confinement at larger distances.

Distributions of individual displacements corresponding to a fixed duration T_{obs} do not immediately support this interpretation. Compared to a normal distribution with the same variance, individual displacements have an excess of short and very long values

(Figure 4.1B). The deviation is more clearly illustrated by the distribution of the individual square displacements (ISD). For Brownian particles, the ISD distribution is exponential, Eq. (2); therefore plots of log $P(r^2)$ should be linear, with slope $-1/(4D \cdot T_{obs})$. Experimentally (Figure 4.1C) the log-ISD distribution displays an upward curvature. This may be interpreted as an effective diffusion constant that gradually increases with distance, in apparent contradiction with the MSD time dependence described above.

Sub-linear time dependence of the MSD of membrane bound molecules typically reflects the presence of obstacles that limit their random movement. Corrals resulting from elements of the cytoskeleton (Niehaus et al, 2008; Sako & Kusumi, 1994; Saxton, 1995), small confining domains consistent with lipid rafts (Simson et al, 1998), or localized trapping sites (Saxton, 2007) may all limit movement at larger distances, resulting in relatively higher mobility at short distances. The reduced short distance mobility seen in the ISD distributions is puzzling; a possible explanation is that the observation time T_{obs} is too large compared to the typical time it takes the particles to cross an obstacle (Kusumi et al, 2005a); but then subdiffusion should also be missed by the MSD time dependence.

4.4.2 The anomalous features of experimental ISD distributions are reproduced by Brownian simulations in a landscape of attractive domains, but not in a landscape of corrals

In order to understand the observed behavior, we performed simulations of two-dimensional Brownian motion in the presence of semi-permeable barriers. Particle movement was simulated using a fixed time step Brownian motion algorithm with

periodic boundary conditions; barriers allowed or blocked individual crossing attempts by particles in a probabilistic fashion, based on a pre-defined permeability, $p_{cross} \leq 1$.

We used two types of barrier configurations, corrals and confining domains. Corrals (Figure 4.2A, inset) represent a cytoskeleton induced partition of the membrane into domains that have similar physico-chemical properties; in our simulations they are separated by boundaries of uniform permeability p_{cross}, that does not vary between domains or with the direction of particles crossing it. Confining domains (Figure 4.2B, inset) represent lipid rafts that have an affinity for the membrane proteins of interest. Simulated confining domain cover a few percent of the membrane surface; their boundaries are fully permeable to particles entering and have permeability $p_{cross} \approx 1 -$ 5% to particles exiting the domain.

First, we performed simulations in a corral landscape, with barriers evenly spaced at distance B = 1. Keeping the same permeability, we varied the observation time T_{obs} so that the corresponding diffusion length $\sigma = 2(DT_{obs})^{1/2}$ probed a range of values around B. All resulting log-ISD distributions (Figure 4.2A) exhibit the expected signature of confinement, in that the apparent diffusion coefficient decreases as r^2 increases toward the barrier spacing $B^2 = 1$. However, the apparent diffusion constant increases again at $r^2 > B^2$ and the alternating pattern is repeated around $r^2 \approx 4B^2$. Also as expected, the confinement signature weakens as T_{obs} increases. For σ values above B, the slowdown at $r^2 \approx B^2$ is increasingly overshadowed by the higher mobility regime $B^2 < r^2 < 4B^2$, and by the second reduced mobility interval at $r^2 \approx 4B^2$. In summary, the log-ISD distributions generated in the corral configuration could explain the absence of higher mobility at low

distances, but none of them exhibited the upward curvature in Figure 4.1C, consistent with higher mobility at larger distances.

This prompted us to consider the confining-domain configuration. The log-ISD distributions in Figure 4.2B were generated using the same algorithm, with permeability and T_{obs} values as the set in Figure 4.2A, but with a single, square shaped attractive domain of size B^2 in the simulation area. Distributions with longer T_{obs} (such that $\sigma > B$) have consistently increased mobility for $r^2 > B^2$. The signature of confinement is also present, with a noticeable but transient decrease in mobility for $r^2 \leq B^2$. For smaller σ values, both features are less pronounced; still, the large distance mobility (or effective diffusion coefficient) is higher than for the corresponding distributions for corrals.

The network of barriers constituted a uniform periodic grid in the corral simulations of Figure 4.2A, and the attractive domain used for Figure 4.2B was a unit square. To eliminate possible artifacts due to the regular geometry, we repeated the simulations in landscapes where the barrier spacing were taken from a random distribution, centered on the same average spacing of $B = 1$; in the corresponding attractive-domain simulations, the attractive domain was one of the rectangles from the corral geometry. We used the same permeability of 1% in the two types of geometries. The resulting ISD distributions (Figure 4.3A-D) are smoother, without pronounced slope variations at $r^2 = B^2$ and $4B^2$. They confirm that the corral geometry does not result in consistently increased long distance mobility, while the attractive-domain geometry always does. In both types of simulation, the MSD versus observation time curve had a decreasing slope. For small T_{obs}, the effective diffusion constant (Figure 4.3F) starts close to the value ($D_0 = 100$

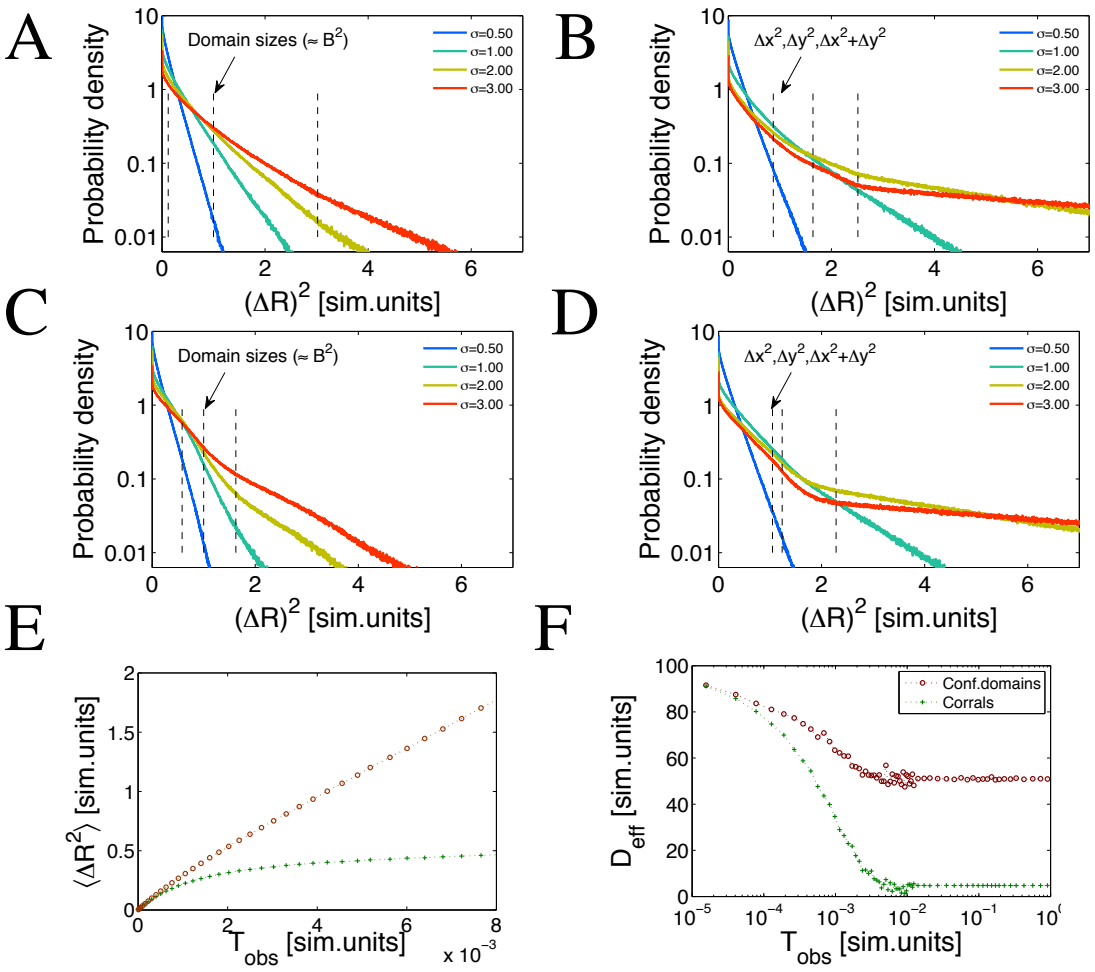

Figure 4.3: Simulations in corral and confining domain landscapes with non-uniform barrier spacing. A-D. Individual square displacement (ISD) distributions in corral (AC) and confining domain (BD) simulations, performed in two different barrier landscapes (AB and CD) with non-uniform barrier distances. The confining domain in B is one of the corral domains from A and the one in D is one of the domains from C. Both types of simulations had the same permability value of 1%. Dashed lines indicate the range and mean of corral spacings (AC), respectively the size of the attractive domain (BD). Variable barrier spacings eliminate the pronounced slope variations from the corral simulations. Unlike the confining domain results, the corral simulations do not exhibit an overall increasing slope. The confining domain simulations are sensitive to the geometry of the domain (which is closer to a square in B and is more elongated in D). E-F. The dependence of the mean square displacement R_2 on the observation time (E) exhibits a decreasing slope in both types of simulations. The decreasing slope is more pronounced in the corral geometry. At short observation times, both simulations are consistent with pure Brownian diffusion (F) and the effective diffusion constant D_{eff} (derived from the slope of the MSD versus time curves) approaches the microscopic value of 100 (simulation units).

sim.units) corresponding to pure Brownian motion, decreases, then remains constant for $T_{obs} > 0.01$.

4.4.3 Experimental displacement statistics are closely approximated by simulations in a landscape of partially confining domains 30-250 nm in diameter

Next, we wanted to see whether the observed ISD and MSD distributions can be approximated by Brownian motion in the presence of a system of barriers. To avoid the artifacts of a periodic barrier structure, we defined a network of unevenly spaced, rectangular barriers (Figure 4.2C).

Compared to the SPT data, simulations in this geometry using the corral configuration (uniformly semi-permeable barriers) exhibited weaker deviations from the ideal Brownian behavior. We obtained a consistently better match to the data with simulations using the attractive-domain configuration, where only a subset of the rectangular compartments were surrounded by barriers, which only impeded exit from the respective domains. We performed simulations with a range of permeability and diffusion length values, in order to match as closely as possible the experimental distributions from one of the three data sets. Direct comparison with the experimental results for the fixed time step distributions reveals a strong qualitative similarity, as illustrated in Figure 4.4A and C. The simulations shown in Figure 4.4 were obtained using a permeability of p_{cross} = 5% and $\sigma = \sqrt{2}B$. The barrier coordinates are $X^B \equiv \{0, 2.70, 6.56, 7.01, 10.00\} \times B$ and $Y^B \equiv \{0, 3.16, 4.50, 6.81, 10.00\} \times B$, resulting in ratios of $\sigma/\Delta X$ and $\sigma/\Delta Y$ in the range of $(0.44 \ldots 3.14)$.

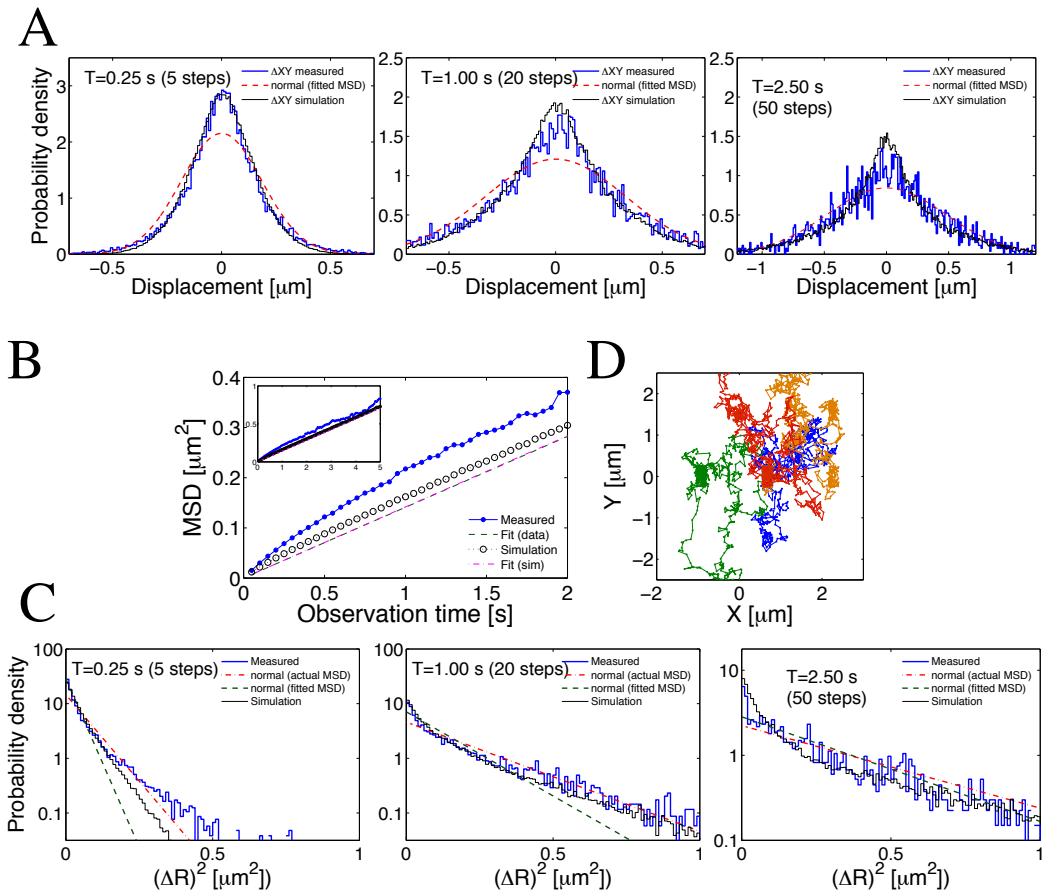

Figure 4.4: Comparison between experiment and simulations. Random sized, trapping (attractive) domains lead to step size distributions that are qualitatively similar to those observed experimentally. A. Distributions of simulated displacements at fixed time closely approximate the experimental result, with a larger discrepancy for small times. B. MSD versus observation time. We used the slope (apparent diffusion constant) of this dependence to convert from simulation to physical units. The discrepancy in the fixed duration distributions (A. and C.) is really due to the differences between the time dependence of the mean square displacement (MSD). The simulation exhibits a smaller vertical offset than the experimental data. C. The distribution of square displacements at fixed time allows for a better comparison. The simulation matches the data better for intermediate observation times. For short times, the simulation underestimates the number of long jumps. D. Simulated trajectories. One aspect that could be improved is the simple geometric shapes we used in these simulations, which should better reproduce the likely varied size and contorted shape of the actual domains (compare with trajectories shown in Figure A.3).

The simulation results were converted to physical units by identifying the observation time with 50 ms and setting the length scale so that the long-term slope of the MSD versus time curve (Figure 4.4B) matched the experimental slope of 0.132 $\mu m^2/s$ (effective diffusion constant of D_{eff} = 0.033 $\mu m^2/s$). This corresponds to B = 77.2 nm, giving barrier sizes of ΔX = {208, 298, 35, 231} nm, ΔY = {244, 103, 178, 246} nm. The highlighted values correspond to attractive domains (Figure 4.2C), whose diameters thus ranged from 35 to 246 nm.

There are discrepancies between the simulated and experimental statistics. Simulated ISD distributions at longer observation time over-estimate the number of small steps. While the simulated MSD distribution (Figure 4.4B) does exhibit sub-diffusion, the initial slope of the experimental curve (D'_{eff} = 0.058 $\mu m^2/s$) is not matched by the simulation. It is interesting to note that the microscopic diffusion constant used in the simulation (D_0 = 0.059 $\mu m^2/s$) is very close to this value, but results in a smaller vertical offset.

4.5 DISCUSSION

The particle tracking data discussed here exhibits two types of deviations from Brownian diffusion, sublinear dependence of the mean square displacement (MSD) on the observation time, and slow apparent diffusion at short distances exhibited by individual square displacements (ISD) at fixed time. Subdiffusion has long been recognized as a likely consequence of obstacles to free movement, such as a network of domains separated by semi-permeable barriers (Kusumi et al, 2005b; Saxton, 1995), or transient trapping (Saxton, 2008). The peculiar features of ISD distributions have been

noted previously, most recently in a careful statistical analysis of FcεRI mobility data (Espinoza et al, 2012).

The present analysis provides a plausible explanation of both features, based on the hypothesis of a collection of small, partially confining domains that occupy a fraction of the membrane area. We propose a model of anomalous diffusion that reconciles the original hop-diffusion model (Kusumi et al, 2005b) with the inhomogeneous, frequently clustered distribution of receptors (Lillemeier et al, 2006).

In the original hop-diffusion model, the membrane is divided (partitioned) into microdomains, mostly by elements of the cytoskeleton that act as semi-permeable barriers to the movement of receptors. Receptors perform Brownian motion with diffusion coefficient D_0 within microdomains, but their movement between domains is limited, except for rare "hop" events. As a result, free diffusion is limited by the typical domain size B. It should be observable in SPT experiments with a high enough frame rate (so that $T_{obs} << B^2/4D_0$). Displacements over distances $r > B$ are unlikely, and the mean displacements over time scales that exceed $B^2/4D_0$ are bounded or increase consistently with a smaller, effective diffusion constant $D_{eff} < D_0$. By the same logic, individual square displacement (ISD) distributions should exhibit decreasing mobility for distances exceeding the diffusion length $\sigma = (4D_0T_{obs})^{1/2}$. We found that this picture was consistent with the experimentally derived MSD curves (Figure 4.1A), but not with the individual square displacement (ISD) distributions, which exhibited increased mobility at higher distances.

The experimental ISD distributions (Figure 4.1C and Figure 4.4C) are likely affected by the relatively long observation time (50 ms), which corresponds to a diffusion length

of ≈ 100 nm. To better understand the implications of the hop-diffusion model, we performed simulations of diffusion in a network of domains (corrals) separated by semi-permeable barriers. The resulting ISD distributions exhibited decreasing mobility only for shorter observation times, indicating a more complex behavior than described above. However, they did not reproduce the monotonically increasing mobility observed experimentally. By contrast, this feature was clearly matched by simulations where all boundaries were fully permeable, except for only one of the domains, which had semi-permeable boundaries in the outgoing direction (Figure 4.2). This led us to formulate a modified version of hop diffusion.

In our proposed model, the microdomains of interest are small, and occupy in aggregate only a fraction of the membrane area; they are partially confining, in that their boundaries are more easily crossed inbound than outbound. By contrast, the rest of the membrane is non-confining and allows high[er] mobility. This model has been implicitly used in (Hsieh et al, 2010; Hsieh et al, 2008; Pryor et al, 2013), in Monte Carlo simulations of signal initiation; however, to our knowledge, it has not been rigorously evaluated for faithful reproduction of receptor mobility characteristics.

The confining domain hypothesis can explain both of the non-Brownian features observed experimentally. Receptors will spend most of their time in a confining domain; rare “hop” events will place a receptor in the open area where it can diffuse quickly until it is trapped by another confining domain. The distribution of individual square displacements (at fixed observation time) results from the overlap of two populations, confined particles with limited movement and escaped particles that diffuse quickly. The former dominate short displacements and the latter dominate long displacements, hence

the steep slope (low mobility) at short distances and lower slope (high mobility) at large distances. Simulations also reproduce the sublinear dependence of the mean square displacement. We performed simulations in a landscape of rectangular shaped domains as described above, in a variety of geometries and mobilities, and were able to obtain a good match to the experimental distributions (Figure 4.4) over a range of observation times.

Similarly to the original hop-diffusion model, unimpeded Brownian behavior is recovered in our model at short observation times (equivalent to a high frame rate). The upward curved shape of the logarithmic individual squared displacement (ISD) distributions and the decreasing slope in the MSD time dependence are robust features of the model; however, the precise shape of both sets of curves is sensitive to the geometry of the barriers, especially to the size and diameter of the attractive domains. Thus, it is very likely that closer matching simulations can be obtained by using varying sizes for the domain barriers, a combination of permeabilities, and irregular shaped domains. This effort should be guided by a theoretically motivated procedure to extract the characteristic parameters from the experimental results.

4.6 NOTES

This work was submitted to Biophysical Journal to be reviewed for publication on March 25th, 2014.

4.7 ACKNOWLEDGEMENTS

This work was supported by National Institutes of Health (NIH) grants R01 GM104973 (to J.S.E. and A.M.H.), K25 CA131558 (to A.M.H), R01 CA119232 and P50 GM085273 (to B.S.W.), and National Science Foundation CAREER award MCB-

0845062 (to D.S.L.). M.M.P. was supported by the National Science Foundation INCBN IGERT Fellowship (DGE-0549500).

CHAPTER 5: ORCHESTRATION OF ERBB3 SIGNALING THROUGH HETERO AND HOMO-INTERACTIONS

Meghan McCabe Pryor,[a,h]* Mara P. Steinkamp,[b,c]* Adam M. Halasz,[d] Ye Chen,[d] Shujie Yang,[e] Marilyn S. Smith[f] Diane S. Lidke,[b,c] Jeremy S. Edwards[a,c,g]^ and Bridget S. Wilson[b,c]^

[a]Department of Chemical Engineering, University of New Mexico, Albuquerque, New Mexico, 87131 [b]Department of Pathology, University of New Mexico Health Sciences Center, Albuquerque, New Mexico, 87131 [c]Cancer Center, University of New Mexico Health Sciences Center, Albuquerque, New Mexico, 87131 [d]Department of Mathematics, West Virginia University, Morgantown, West Virginia, [e]Department of OB/GYN, University of Iowa Carver College of Medicine, Iowa City, Iowa [f]Viracor-IBT Laboratories, Lee's Summit, Missouri 64086 [g]Department of Chemistry & Chemical Biology, University of New Mexico, Albuquerque, New Mexico, 87131 [h]Center for Nonlinear Studies, Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545,

*Contributed equally to this work

^Communicating authors. Address correspondence to: bwilson@salud.unm.edu or jedwards@salud.unm.edu

5.1 ABSTRACT

Members of the erbB family of receptor tyrosine kinases are capable of both homo- and hetero-interactions. Because each receptor has a unique set of binding sites for downstream signaling partners and differential catalytic activity, subtle shifts in their combinatorial interplay may have a large impact on signaling outcomes. Overexpression and mutation of erbB family members are common in numerous human cancers and alter the balance of activation within the signaling network. Here we report the development of a spatial stochastic model that addresses the impact of varying ligand concentrations and erbB2-erbB3 ratios on erbB3 homo and hetero-interaction dynamics and phosphorylation state. We also report experimental and computational analysis of an erbB3 gain-of-function mutant, located in the C-lobe asymmetric dimerization interface, which shows enhanced phosphorylation at low ligand dose associated with increased kinase activity.

5.2 INTRODUCTION

The ErbB family of receptor tyrosine kinases consists of four related receptors that form both homo- and heterodimers (Lemmon & Schlessinger, 2010), as well as potentially higher-order oligomers (Kozer et al, 2013). This work focuses on the unusual properties of erbB3, that are markedly dependent upon engagement with a heterodimerizing partner for its transphosphorylation and upregulation of its inherently weak catalytic activity (Shi et al, 2010; Steinkamp et al, 2014). Previous studies have suggested that erbB3 favors erbB2 over other erbB family members for heterodimerization (Zhang et al, 2009) and erbB2/erbB3 interactions are important in both normal developmental processes (Yarden & Sliwkowski, 2001), as well as cancer initiation and progression (Liu et al, 2007; Vaught et al, 2012). Although erbB3 and

erbB4 share heregulin/neuregulin 1 as a ligand (HRG/NRG1) and can form heterodimers (Monsey et al, 2010), it is only the erbB3/erbB2 complex that is implicated in melanoma growth and survival (Zhang et al, 2013). ErbB3 expression has been linked to resistance of tumors to tyrosine kinase inhibitor therapies (Huang et al, 2012; Lee et al, 2014; Sato et al, 2013; Sergina et al, 2007), motivating more comprehensive analyses of erbB3 dynamics and signaling capabilities.

Mathematical modeling has emerged as a powerful method to explore the complexity of erbB family signaling (Andrews & Bray, 2004; Blinov et al, 2006; Kholodenko et al, 2010; Kleiman et al, 2011; Radhakrishnan, 2010; Zhang et al, 2009), particularly when supported by high quality, quantitative measurements at different time and length scales (Pryor et al, 2013; Shankaran et al, 2013). Most prior computational studies have neglected erbB3 as a significant catalytic entity. A notable exception is the work of Telesco et al. (2011), who suggested that even the minimal level of basal erbB3 autophosphorylation could alter the dynamics of erbB signaling pathways and contribute to drug resistance. Recently, we demonstrated that tyrosine kinase activity is significant in immuno-isolated, intact erbB3 after binding heregulin (HRG), providing that erbB2 is co-expressed (Steinkamp et al, 2014). High resolution imaging methods, such as single particle tracking (SPT), captured erbB3-erbB3 and erbB3-erbB2 interactions in real time and revealed that ligand-bound erbB3 engaged in homointeractions that were 3-4x more stable than when bound to erbB2 (Steinkamp et al, 2014). These observations raised the intriguing questions: Under what conditions might erbB3's catalytic activity be important? And, further, what is the potential impact of spatial organization and serial engagements of both homo- and heterodimer configurations upon signaling?

We approach these questions through a stimulation platform designed to reflect characteristics of the membrane landscape, that underlie the anomalous diffusion characteristics of erbB receptors (Low-Nam et al, 2011; Steinkamp et al, 2014) as well as the random nature of receptor encounters in the 2D fluidic membrane. Recently applied to the study of erbB1/EGFR homodimerization (Pryor et al, 2013), our spatial stochastic simulation method incorporates a Rule-Based approach that explicitly considers important structural features that control erbB signaling. These features include the stable upright configuration of the extracellular domain of erbB2 or ligand-bound erbB3, rendering them dimerization competent throughout the simulation. In contrast, the unstable flux of resting erbB3 from the bent state to a dimerization-capable state is represented by a probability term; at each microsecond time step in the simulation only 1% of resting erbB3 receptors are upright and available to dimerize. The model also assumes that the asymmetrical orientation of kinase domains in each dimer is a random process, such that each monomer in the pair is assigned as a *receiver* or an *activator* respectively. This simulation strategy sets up a scenario where individual receptor monomers swap their status as a receiver or activator exclusively through a stochastic process of dissociation and rebinding. During these encounters, the model tracks the phosphorylation state of each receptor monomer, adding an experimentally estimated multiplier for increased catalytic activity so long as the monomer remains phosphorylated. The simulation is governed by experimentally-defined rate constants for phosphorylation, dephosphorylation, and dimer dissociation. ErbB receptors cycle through many encounters during these simulations, revealing properties that drive signal

propagation and pointing to important influences such as the relative ratios of each receptor species, density, and dwell times.

A novel addition to the model is the use of single particle tracking data to estimate the size of membrane domains ("confinement zones") (Simson et al, 1995) that transiently trap erbB2 and erbB3, incorporating previously described overlap of these domains (Hsieh et al, 2008) with their respective dwell times. These factors translate in the model to probabilities for escape from confinement. Finally, we utilize the model to analyze experimental data collected in cells expressing a novel gain-of-function mutation in erbB3. Located within the C-lobe of the erbB3 kinase domain, the E933Q mutation was initially discovered by us as a heterozygous mutation in the SKBR3 breast cancer cell line. We show that erbB3^{E933Q} expressed in CHO cells is more readily phosphorylated at low ligand dose than wildtype erbB3 (erbB3^{WT}), with accompanying higher catalytic activity. Since SPT data showed that the gain-of-function status was not linked to longer homodimer lifetimes for erbB3^{E933Q} compared to erbB3^{WT} and ligand binding was unaltered, the model was utilized to provide an estimate of the mutant's increased catalytic activity. This work sets the stage for mechanistic profiling of the entire spectrum of erbB3 oncogenic mutations (Jaiswal et al, 2013), which are distributed across both the extracellular and kinase domains and may each have unique contributions to dimer stability or kinase activity.

5.3 RESULTS

5.3.1 ErbB3 dephosphorylation shows a lag after acute inhibition of erbB2 catalytic activity

Ligand-dependent erbB3 phosphorylation requires erbB2, since it can be abrogated by pretreatment with pertuzumab or lapatinib (Steinkamp et al, 2014; Zhang et al, 2009). However, it is not known whether erbB2 activity is also required to *maintain* erbB3 phosphorylation after stimulation has upregulated erbB3 activity. Figure 5.1AB shows results obtained in CHO cells expressing HA-tagged erbB2 ($erbB^{HA}$) and GFP-tagged erbB3 ($erbB^{GFP}$). Cells were serum starved and stimulated for short duration with 12 nM heregulin β (HRG), followed by treatment with lapatinib to acutely inhibit erbB2 kinase activity. Lysates were collected at defined time intervals after adding lapatinib and western blots were probed with phospho-specific antibodies against erbB3 tyrosine 1280 (PY1289) or erbB2 tyrosine 1248 (PY1248) to measure receptor dephosphorylation over time. Results show that HRG stimulation alone led to elevated levels of PY1289 on erbB3, which were maintained for up to 30 minutes after stimulation (Figure 5.1A). The plot in Figure 5.1B (blue line) shows that addition of 10 μM lapatinib caused rapid dephosphorylation of erbB2 PY1248 with a half-life (t ½) of ~14 seconds, comparable to reported values (Kleiman et al, 2011). The drop in erbB3 phosphorylation at PY1289 demonstrated a lag, with a half-life of ~3.5 minutes (orange line, Figure 5.1B).

A potential explanation for this lag is that erbB3 phosphorylation is sustained for short periods after erbB2 shutdown through its own improved autophosphorylation capacity. To evaluate this, we developed a Rule-based non-spatial model using BioNetGen (Faeder et al, 2009), that incorporated erbB3/erbB2 heterodimer and erbB3/erbB3 homodimer lifetimes determined by SPT (Steinkamp et al, 2014) and equal expression levels of both receptors, with 50% ligand bound erbB3. These and other parameters for our dimerization models are listed in Table 4, along with sources for each

Species	ErbB2	ErbB3	ErbB3-EQ	ErbB2. ErbB2	ErbB2. ErbB3	ErbB3. ErbB3	ErbB3EQ. ErbB3EQ
Diffusion Coefficient [um^2/s][1,2]	0.0272	0.0310	0.0621	0.0150	0.0150	0.0185	0.0139
Dimer On Rate [um^3/s][3]	-	-	-	0.0034	0.0034	0.0034	0.0034
No-Ligand Dimer Off Rate [1/s][1,2]	-	-	-	0.4360	0.4360	0.4360	0.436
Phosphorylation Rate [1/s][4,5]	0.0730	0.00008	-	-	-	-	-
Dephos. Rate [1/s][1]	0.0725	0.0725	0.0064	-	-	-	-
Phosphorylation Multiplier [Unitless][1]	2	0.53	varied	-	-	-	-
Ligand Dimer Off Multiplier [Unitless][1]	-	-	-	-	0.9358	0.3	0.30
Phos Diffusion Multiplier [Unitless][1]	-	-	-	1	0.31	0.15	0.48

Table 4: Parameters for ErbB2/3 and erbB2/ErbB3E933Q simulations. 1 – Experimental data in this paper, 2 – Steinkamp et al (2014), 3 – Shankaran et al. (2013), 4 – Kleiman et al (2011), 5 – Shi et al. (2010)

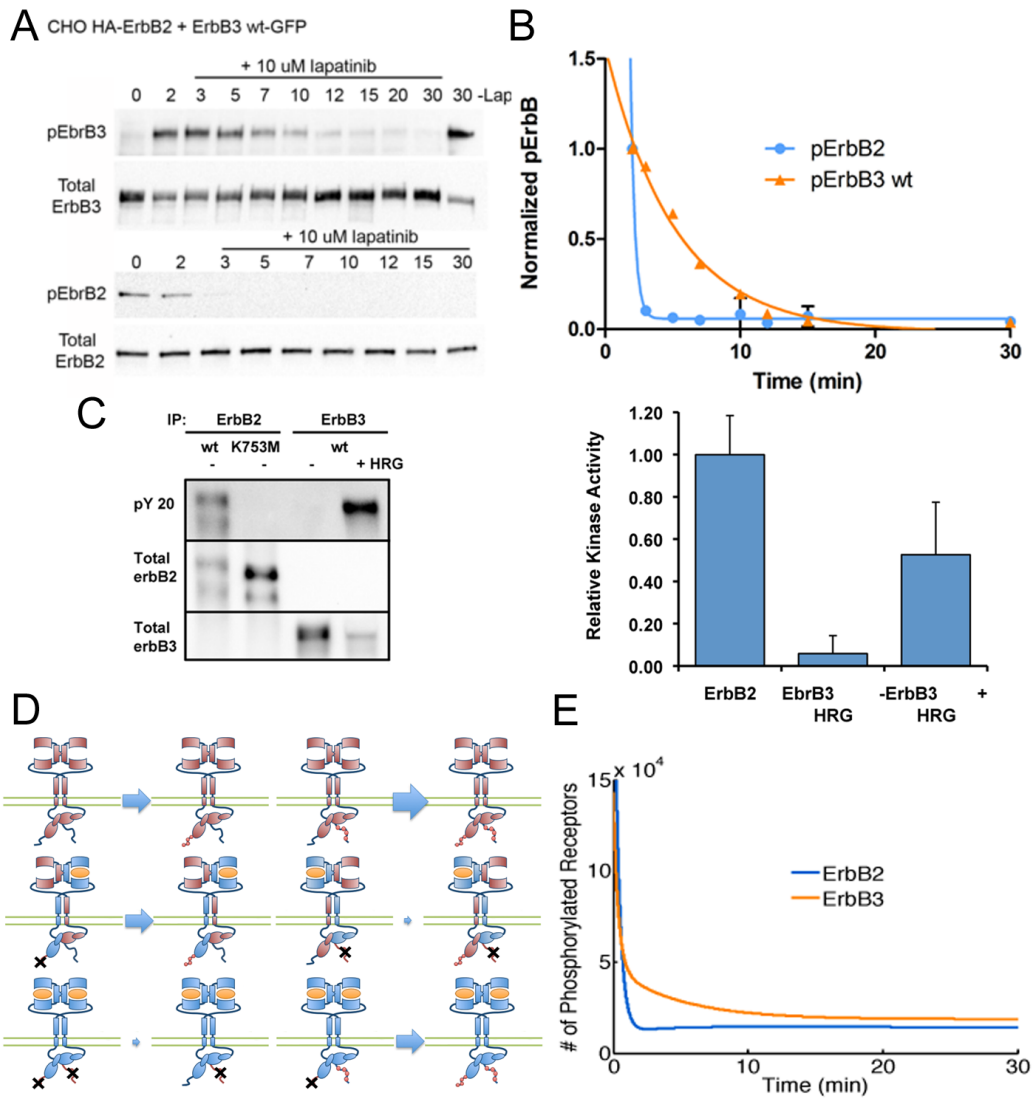

Figure 5.1: ErbB3 kinase activity and dephosphorylation (A) CHO cells expressing exogenous HA-erbB2 and erbB3 were stimulated with 12 nM HRG for 2 minutes at 37°C and then treated with the erbB2 kinase inhibitor, lapatinib. ErbB3 phosphorylated at Y1289 or erbB2 phosphorylated at Y1248 were detected using phospho-specific antibodies. Levels of phospho-receptors were normalized to the total receptor levels. (B) The normalized phosphorylation levels for erbB2 and erbB3 were plotted over time. Phosphorylation levels of both receptors were set to 1 for the two minute time point (maximum HRG stimulation). Points were fit to a one-phase exponential decay curve to determine the dephosphorylation half-life (C) Western blot of immunoprecipitated erbB2 or erbB3 from CHO cells expressing ErbB2-mYFP (left two lanes) or ErbB3-mCitrine (right two lanes). The kinase activity of immunoprecipitated samples was then normalized to the levels of phospho-receptor. Comparison of the relative kinase activity of erbB2 and erbB3 +/- HRG after normalization. ErbB2 kinase levels were set to 1. Data shown is the average of two independent trials +/- STDEV. (D) Phosphorylation reactions. Our model allows phosphorylation to occur between all dimer types. The phosphorylation rate (or kinase activity) is varied based on the dimer type. Unphosphorylated ErbB33 homodimers have an extremely low phosphorylation rate (Lemmon), while phosphorylated ErbB22 homodimers have a high phosphorylation rate (Steinkamp). We employ the phosphorylation shuffle mechanism used in Pryor et al. (2013). (E) Results from the BioNetGen model predicting a similar erbB3 dephosphorylation curve when erbB3 kinase remains active after lapatinib treatment.

value. In the simulation, phosphatase activity against both erbB2 and erbB3 were equivalent. Kinase activity for each monomer is governed by a set of rules, including dimerization competency based on ligand occupancy (for erbB3). ErbB2 kinase activity was based upon published estimates for the basal rate (Kleiman et al, 2011), adjusted by a multiplier when phosphorylated (Shankaran et al, 2006). To estimate a corresponding multiplier value for erbB3, we compared catalytic activity in erbB2 immune complexes versus erbB3 immunoisolated from resting or HRG-stimulated cells (Figure 5.1C); results were normalized for loading based upon immunolabeling with the commercial pan-reactive anti-phosphotyrosine antibody, PY20. In the simulation, lapatinib addition converted erbB2 kinase activity immediately to zero. ErbB3 kinase activity is insensitive to lapatinib (Shi et al, 2010). Therefore, in the simulation, erbB3 kinase activity is set at its multiplier of 0.56 of the phosphorylated erbB2 receptor phosphorylation rate, as long as erbB3 remains phosphorylated.

Figure 5.1D illustrates some of the possible dimer configurations that occur in the simulation, to include homodimers of erbB2 (red-red), erbB3 (blue-blue) and erbB3-erbB2 heterodimers (red-blue). During the simulation, there is a random assignment of the orientation of their cytoplasmic tails such that only one monomer is upregulated through contact of its N-lobe with the C-lobe of the activator. In this cartoon, ligand bound to erbB3 is represented by orange ovals while phosphorylation is indicated by colored features in the cytoplasmic tails. The arrows for each reaction depicted scale according to the phosphorylation rate associated with the dimer state of the specific reaction. Results of the simulation (Figure 5.1E) match well with the experimental results: erbB2 phosphorylation drops off rapidly while erbB3 phosphorylation declines

slower due to encounters between erbB3 monomers that have been phosphorylated (primarily by erbB2) in a prior binding event.

5.3.2 Single particle tracking of erbB3 and erbB2 diffusion indicates that receptors transiently reside in partially overlapping confinement zones

Our spatial stochastic model includes the capability to consider the impact of membrane topography on diffusion-limited reactions between receptors. Our previous models relied on "snap-shot" images of erbB receptor distributions on fixed cell membranes, as observed by immuno-electron microscopy, to populate the 2D simulation landscape with domains that transiently confined receptors. Our next goal was to expand on this concept by using SPT data to estimate the area and shapes of confinement zones, as well as to address the possibility that erbB3 and erbB2 receptors might exhibit individual characteristics such as domain dwell time and distribution. For this portion of the study, we used data sets acquired using 2-color quantum dot (QD) tracking on stably transfected CHO cells, where $erbB2^{HA}$ was tracked with anti-HA-FAB-QD^{585} and erbB3 was tracked with HRG-conjugated QD^{655}. An example of trajectories from a single observation area is shown in Figure 5.2A, where three erbB2 and two erbB3 were tracked simultaneously (defined by color coding in the legend to Figure 5.2A).

To analyze multiple datasets containing 2-color trajectories, we developed and applied a Domain Reconstruction Algorithm (DRA). The DRA converts dynamic trajectories to static spatial data that can be used to approximate the size and contours of confinement zones occupied by erbB3 and erbB2 on the CHO cell membranes. The algorithm is fully described in Chapter 2. In brief, SPT trajectory data is first sorted into two groups that reflect either the confined or freely diffusing state. For each point, a

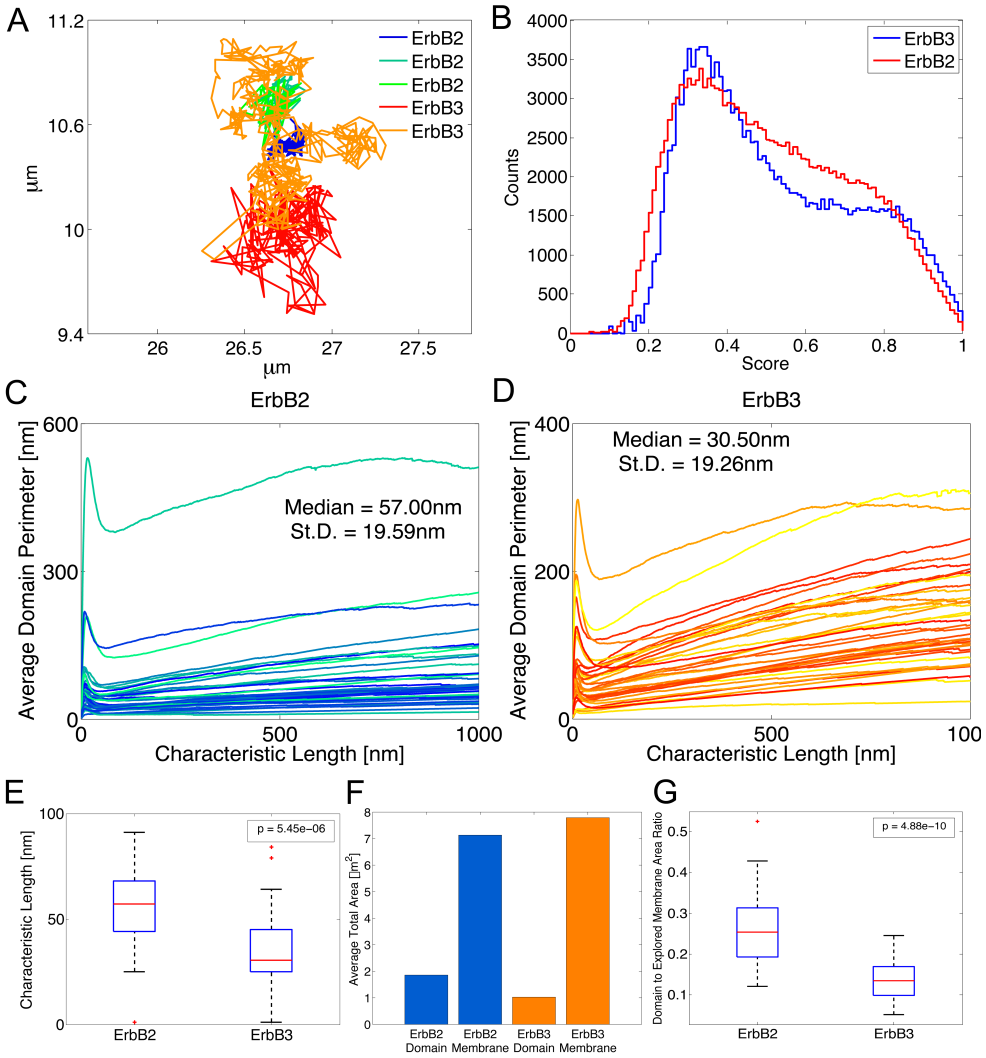

Figure 5.2: (A) Sample of trajectories from SPT data showing overlap of ErbB2 and ErbB3 receptors. This is the initial data form before DRA analysis. (B) Compiled ranks for all the points from the SPT data for ErbB2 and ErbB3. A bimodal distribution is apparent for both species. The local minimum between the two modes was used to estimate the cutoff rank to group the points into long and short points. Compilation of the ranks for SPT experiments where ErbB2 was tracked using QD-585 and ErbB3 was tracked using QD-655. (C and D) Characteristic length study used to determine the characteristic length for clustering. The average domain perimeter was calculated over a range of characteristics lengths. The minimum of the average domain perimeter was used to determine the optimal characteristic length. (C) is the characteristic length study for ErbB2 and (D) is the characteristic length study for ErbB3. (E) Box plots of characteristic lengths calculated during the characteristic length study for each SPT data file. ErbB2 characteristic length is compared to ErbB3. Using the Kruskal-Wallis Nonparametric Test, the characteristic lengths for ErbB2 and ErbB3 are statistically different, p value = 0.0000545. (F) Average total areas of domains and explored membrane for ErbB2 and ErbB3. The explored membrane was calculated using the DRA and setting the characteristic length to the localization error of the SPT experiments. While ErbB2 domains appear to be larger on average, ErbB3 receptors tend to explore more of the open membrane. (G) Box plots of the ratio of domain area to explored membrane area for ErbB2 and ErbB3. Using the Kruskal-Wallis Nonparametric Test to compare between ErbB2 and ErbB3, the ratios are found to be statistically different, p value = $4.88e^{-10}$.

ranking system is applied that compares the preceding and subsequent jump sizes in the trajectories against one another. When these ranks are compiled and sorted into a histogram, a bimodal distribution becomes apparent (Figure 5.2B). The local minimum of the bimodal distribution is then used to determine a cutoff rank to separate the confined points from the freely diffusing points. The analysis was applied to 25 SPT data sets to generate the plots in Figure 5.2B. Results were comparable when applied to 13 SPT data sets where the QD probes were reversed as an important control measure since there is a slight difference in localization accuracy for the two classes of QDs (QD^{655} and QD^{585}; Figure A.6). Based upon these results, a cutoff score of 6.5 was utilized for further DRA analyses.

An estimation of domain size was our next goal, incorporating a clustering algorithm based on work by Espinoza et al. (2012) that assigns a reasonable length parameter as the maximum distance between two points for them to be considered in the same cluster. The average perimeter of all the clusters in a single file is computed over a range of clustering characteristic lengths. This average perimeter is then compared across all the SPT tracking files. Figure 5.2C and Figure 5.2D show the average perimeter of domains for each characteristic length from the ErbB2 tracking data and ErbB3 tracking data, respectively. We defined the average characteristic length of the clusters for each receptor based on the local minimum in these plots. This local minimum corresponds to a minimum of domain perimeters. “Inflating” the perimeter points in each cluster by one-half the characteristic length creates the final confinement shape and area.

Based upon the DRA results, we are able to compare the membrane domain characteristics for ErbB2 and for HRG-bound ErbB3. We used the Kruskal-Wallis

nonparametric test, which does not assume any specific distribution type, only that the data sets being compared come from the same distribution. We find that the characteristic lengths for ErbB2 and ErbB3 domains are statistically different ($p = 0.0000545$). Box plots of the two data sets are shown in Figure 5.2E, confirming that the characteristic lengths for erbB2 and HRG-bound erbB3 clusters on CHO membranes are 57nm and 30nm, respectively. We note that cluster size and total domain area for these receptors may vary in different cell types (Nagy et al, 2010; Yang et al, 2007).

The next important membrane characteristic to take into account is the density of domains on the membrane. A hallmark of SPT experiments is the sparse labeling of receptors to allow for single particle resolution. One method to compensate for this sparse labeling is to estimate the total area of the membrane explored by labeled receptors from a single SPT data file and compare it to the total domain area of the membrane from the same SPT data file. To calculate the area of the membrane explored by a receptor, the DRA was used to analyze the SPT data using a characteristic length equal to the average localization error of the SPT experiment. Computing the ratio of Total Domain Area to the Total Explored Membrane Area allows the sparse SPT domain reconstruction information to be applied to the full cell membrane. As shown in Figure 5.2F,G, the ratio of Total Domain Area to Total Explored Membrane Area can be computed for each SPT data file for each receptor. This ratio can be compared between ErbB2 and ErbB3 to determine whether both receptors explore the same membrane space. Using the Kruskal-Wallis Nonparametric Test, we find that this ratio is also statistically different between ErbB2 and ErbB3 ($p = 4.88 \times 10^{-10}$). The results of these two sets of tests suggest that

motion of ErbB2 and ligand-bound ErbB3 is differentially constrained within the membrane landscape.

5.3.3 The simulation landscape is studded with overlapping domains consistent with experimental results.

Figure 5.3 shows the final simulation space created using the cumulative information from the DRA analyses. The simulation space includes a defined number of ErbB2 and ErbB3 receptors, as well the domain spaces in which they are transiently confined. Density of receptors was determined by calculating the number of receptors per μm^2 of membrane surface in a CHO cell, reported to be 588 μm^2 (Miyagi & Maruyama, 2010). Given our estimates of ~500,000 of each receptor species in these stably transfected cells, we derive a receptor density of ~850 copies of each receptor per 1 μm^2 of membrane – or a total of 1700 receptors/μm^2. During the stochastic simulations, every receptor is accounted for individually. This level of detail in simulations significantly increases the computational costs when running simulations. To reduce the computing time, the simulation space was converted to a total surface area of 0.1 μm^2, equating to ~170 of each type of receptor.

Note that while the domains are statistically different, analysis of the single particle tracking data also revealed an overlap between the two types of domains. This result is consistent with the findings from our prior immuno-electron microscopy study (Yang et al, 2007). Next, the ratio of the Domain Area to Explored Membrane Area ratio was used to determine the amount of domain cover for the desired surface area. The reconstruction of these irregular domains is depicted by blue (erbB2) and orange (erbB3) contours in Figure 5.3. During the simulations, receptor species exhibit no preference to enter their

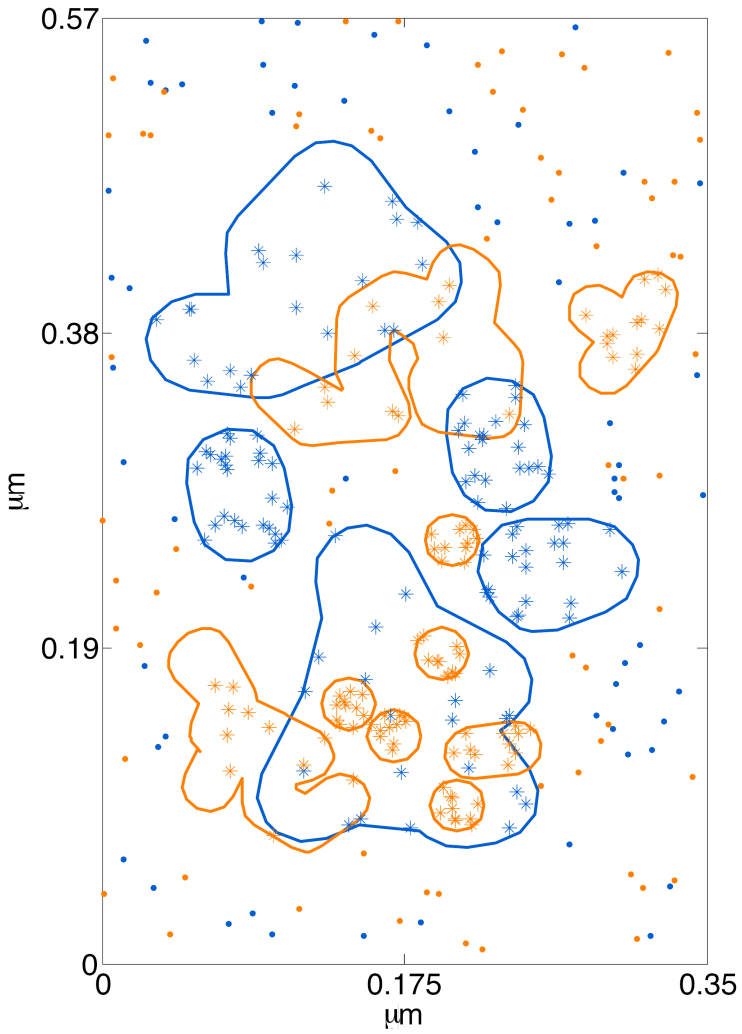

Figure 5.3: Reconstructed Simulation Space (A) Simulation Space Constructed based on the DRA analysis. Domains from a SPT data file were used, and then domains from the same file were added until the domain area to explored membrane area ratio was met. The receptor density was calculated using the approximate number of receptors per cell and average surface area of a CHO cell, then scaling to the simulation space area.

respective domains but a penalty for escape is imposed such that at steady state receptors occupy free and domain spaces according to the DRA cutoff score (65% in domains, 35% unconstrained).

5.3.4 Receptors cycle repeatedly through heterodimer and homodimer reactions.

Results of spatial simulations are shown in Figure 5.4. In Figure 5.4A,B, the relative ratios of erb2 and erbB3 were equal and 50% of the erbB3 were prebound to ligand at the start for simplicity. By illustrating the dimer and phosphorylation events for a single erbB2 monomer (Figure 5.4A) or HRG-bound erbB3 monomer (Figure 5.4B), the plots illustrate the essential stochasticity of the system. The erbB2 in 4A spends time as a monomer, as an erbB2-erbB2 homodimer and as an erbB2-erbB3 heterodimer, with the phosphorylation state of each monomer in the pairs tracked throughout. Similarly, the erbB3 receptor in 4B cycles through monomer, homodimer and heterodimer states. Due to the more stable lifetime, erbB3 homodimers are a predominant feature.

Simulations were also performed varying the ratios of erbB2 to erbB3, as well as the ligand dose. In Figure 5.4CD, a scenario of 20-fold excess erbB3 over erbB2 was considered. Here, the representative plot of an erbB2 shows it to cycle predominantly through heterodimer reactions, with short durations as an unphosphorylated monomer. The plot of one of the erbB3 species in the simulation follows it through at least one binding event with erbB2, leading to dual phosphorylation, followed by relatively stable periods as a homodimer and short periods as a monomer. Finally, the plots in Figure 5.4E,F show results for the case of equal ratios of receptors but a lower ligand occupancy.

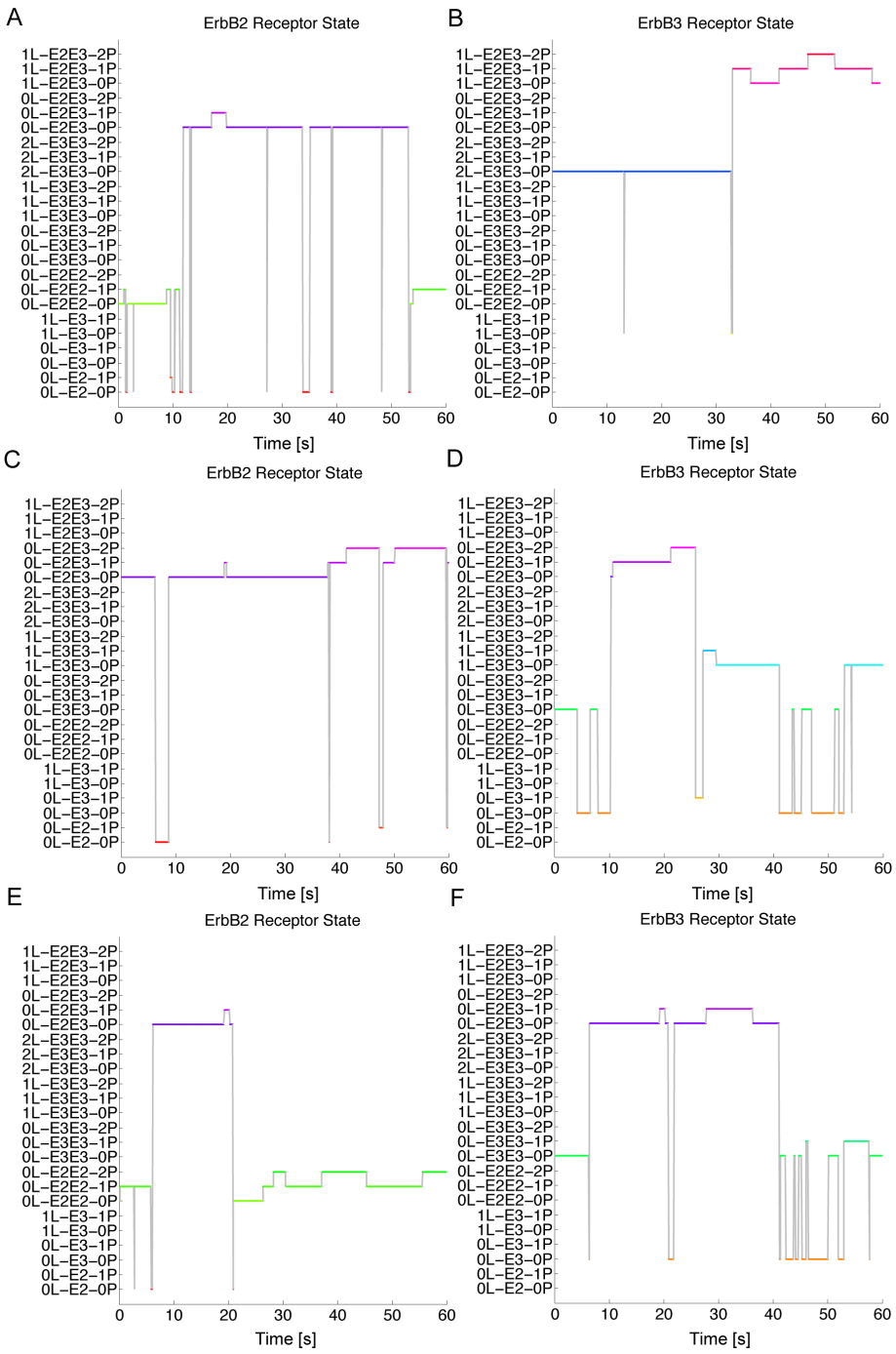

Figure 5.4: Receptor States over time. (A and B) State changes for ErbB2 and ligand bound ErbB3, respectively, for simulations with equal amounts of ErbB2 and ErbB3, and 50% ligand bound ErbB3 present. (C and D) State changes for ErbB2 and unliganded ErbB3, respectively, for simulations with 1/20th the amount of ErbB2 to ErbB3, and 50% ligand bound ErbB3 present. (E and F) State changes for ErbB2 and unliganded ErbB3, respectively, for simulations with equal amounts of ErbB2 and ErbB3, and 15% ligand bound ErbB3 present.

5.3.5 Steady state analysis of erbB phosphorylation reactions underscores the dependency of erbB3 activation on erbB2 and a potentially significant role for erbB3 kinase

Figure 5.5 presents summary data for a series of spatial stochastic simulations, varying conditions such as receptor ratio and dose. The graphs report the total phosphorylation state of each species and the kinase species responsible for the transphosphorylation reactions. As shown in Figure 5.5A, when erbB3 is the only species in the simulation (total 170), there is no phosphorylation. When the simulation parameters are set to equal erbB3-erbB2 levels (total 340), measurable erbB3 phosphorylation results after encounters between erbB3 that "flux" to the upright position and bind erbB2. The number of ErbB3 phosphorylation events at 15% and 50% ligand occupancy are 40 and 45, respectively. At 100% ligand occupancy, the number of phosphorylation events involving an ErbB3 receptor nearly doubles, to 77 events.

Results in Figure 5.5BC show the relative contributions of each receptor for transphosphorylation of the same species (while in homodimers) or of the opposite species (while in heterodimers) is a dose-dependent. For example, Figure 5.5B shows erbB2 homointeractions drive most of the erbB2 and erbB3 phosphorylation events when there are equal numbers of both receptors. By comparison, erbB3's catalytic activity mediates 10-24% of erbB2 phosphorylation events and less than 5% of the erbB3 phosphorylation events. Figure 5.5C reports results when erbB3 outnumber erbB2 20-fold. Here erbB2 densities are low and erbB homodimerization/transphorylation events are infrequent. However, erbB2's phosphorylation of erbB3 is a critical step, since it

governs the erbB3 activation step. ErbB3 receptors do not start phosphorylating other receptors until they

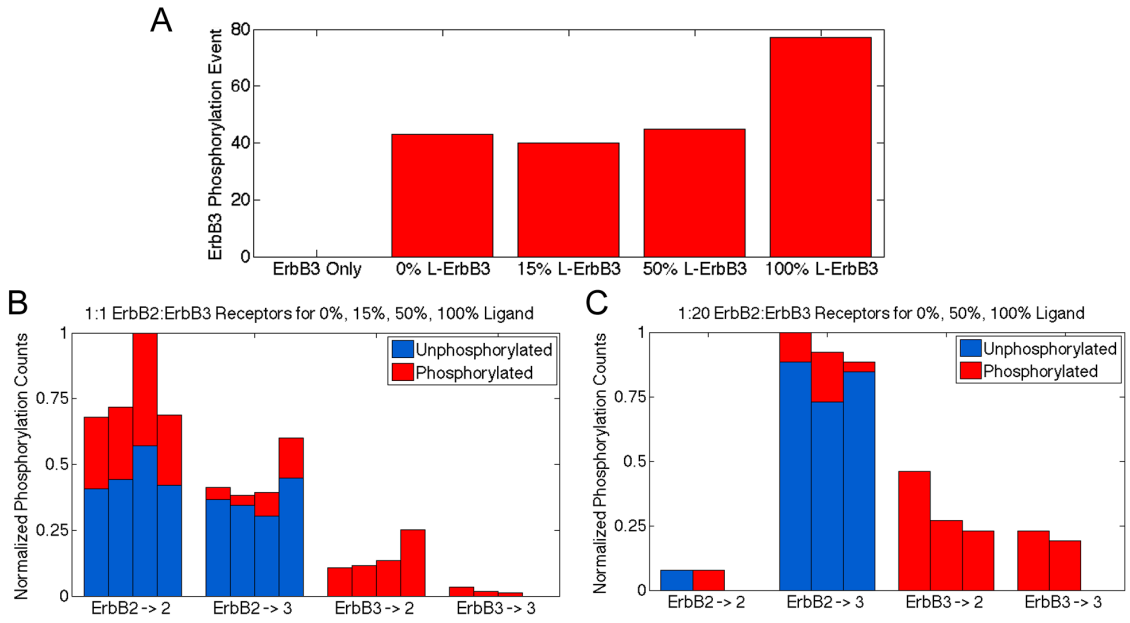

Figure 5.5: Spatial Stochastic Simulation Results. (A) ErbB3 phosphorylation events. When ErbB2 is not present, ErbB3 receptors fail to phosphorylate other ErbB3 receptors. As the amount of liganded ErbB3 increases, the number of ErbB3 phosphorylation events also increases. (B and C) Phosphorylation events by activator species for varying amounts of liganded ErbB3. (B) Equal amounts of ErbB2 and ErbB3. Each bar in a group represents a ligand percentage for ErbB3, 0%, 15%, 50%, and 100% respectively. The majority of ErbB3 phosphorylation comes from ErbB2. ErbB3 receptors only phosphorylate other receptors once phosphorylated themselves. (C) Low amount of ErbB2 receptor (5%) compared to ErbB3 receptors. Each bar in a group represents a ligand percentage for ErbB3, 0%, 50%, and 100% respectively. Majority of ErbB3 phosphorylation still comes form ErbB2, however there is higher activity in ErbB3 homodimers.

have been phosphorylated first. Activated erbB3 is responsible for the large majority of phosphorylation events where there is 0 or 50% ligand occupancy. Note that, at 100% ligand occupancy, erbB3-erbB3 transphosphorylation is not significant. This is explained by the long lifetime – and the likelihood that repeated interactions of erbB3 pairs in homodimers renders it less likely for a productive "activating" dimerization event with erbB2.

Note that elevated basal levels of erbB2 phosphorylation were noted previously in cultured cells overexpressing erbB2, along with a slight reduction upon stimulation with HRG (Steinkamp et al., 2014; Yang et al., 2007). Thus the simulation results for erbB2 phosphorylation point to a high degree of erbB2 homodimerization and activation at high surface density and are in good agreement with experimental results.

5.3.6 Computational approaches offer tractable methods to evaluate impact of mutations on erbB3-mediated signaling.

We next extended out model to look into how mutations that impact erbB3 activation would impact these dynamics. To replicate a mutation in erbB3 that causes the kinase domain to become more active, we added another multiplier to the model. This multiplier is applied on top of the experimentally determined multipliers for erbB2 and erbB3. In the simulation, when a mutant erbB3 is chosen to be an activator, the multiplier is applied to the phosphorylation rate. This simulates an increase in activity to the C-lobe of the receptor.

In Figure 5.6AB, the impact of the multiplier on the percent phosphorylation of erbB2 and erbB3 over time is shown. These simulations used equal amounts of erbB2 and erbB3, with a low amount of ligand-bound erbB3 present (15%). The simulated mutation

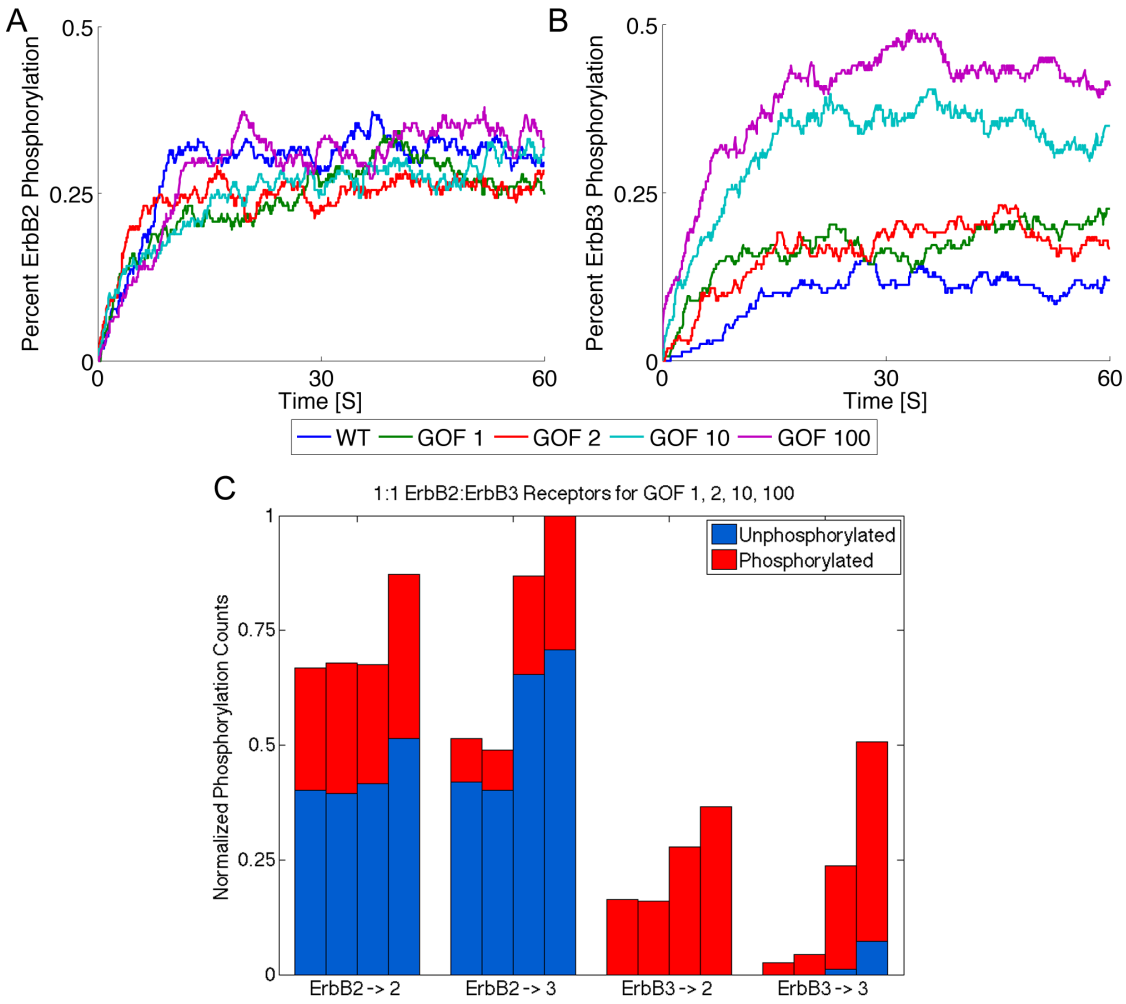

Figure 5.6: Gain of Function Impact on Receptor Phosphorylation for Low Amounts of Ligand. (A and B) Percent receptor phosphorylation over time for ErbB2 and ErbB3, respectively. (A) Initially the GOF mutation has a negative impact on ErbB2 phosphorylation. Overall, the GOF mutation does not show a large impact on the amount of ErbB2 phosphorylation. (B) ErbB3 phosphorylation increases slightly with a modest increase in phosphorylation rate multiplier, and jumps drastically with a multiplier of 10. (C) Phosphorylation events by activator species for increasing phosphorylation rate multipliers. Each bar in a group is a specific multiplier, 1, 2, 10, and 100, respectively. Each group indicates what species the receiver was and what species the activator was for a specific phosphorylation event. When the multiplier reaches 10, ErbB33 homodimers begin to be active without needing to interact with ErbB2 first.

appears to have little impact on erbB2 phosphorylation, however erbB3 phosphorylation is greatly impacted. Figure 5.6C shows the relative distribution of phosphorylation events broken into groups based on which receptor was the activator and which receptor was the receiver. Each bar in a group is for a multiplier of 1, 2, 10, and 100, respectively. Overall, this figure shows that erbB phosphorylation events increase with the multiplier, as would be expected. An interesting observation highlighted by this figure is that when the multiplier reaches 10, erbB3 begins to overcome its reliance on erbB2 for activation, as indicated by the unphosphorylated erbB3 contributing to the phosphorylation of other erbB3 receptors.

5.3.7 Substitution at erbB3-E933Q is a gain-of-function mutation that amplifies the PI3 kinase/AKTsignaling pathway.

We discovered a novel amino acid substitution mutation in one of the erbB3 alleles in the widely-used SKBR3 breast cancer cell line (Figure A.7). To evaluate the possibility that this was an activating mutation, CHO cells were transfected with vectors for expression of mCitrine-fusion proteins bearing either erbB3^{WT} or erbB^{E933Q}. After selection to ensure comparable surface expression of the two fluorescent erbB3-fusion proteins, cells were challenged with a series of HRG doses, lysed, and subjected to SDS-PAGE and western blotting to measure levels of erbB3 phosphorylation at PY1289. Robust phosphorylation of mutant erbB3 was consistently detected at the lowest dose of HRG, with *2 to 3 fold stronger* phosphorylation than wild type erbB3 at suboptimal ligand concentrations (Figure 5.7A).

We reasoned that the robust phosphorylation of ErbB3^{E933Q} in response to low doses of ligand might amplify downstream signaling by increased recruitment of Class I A

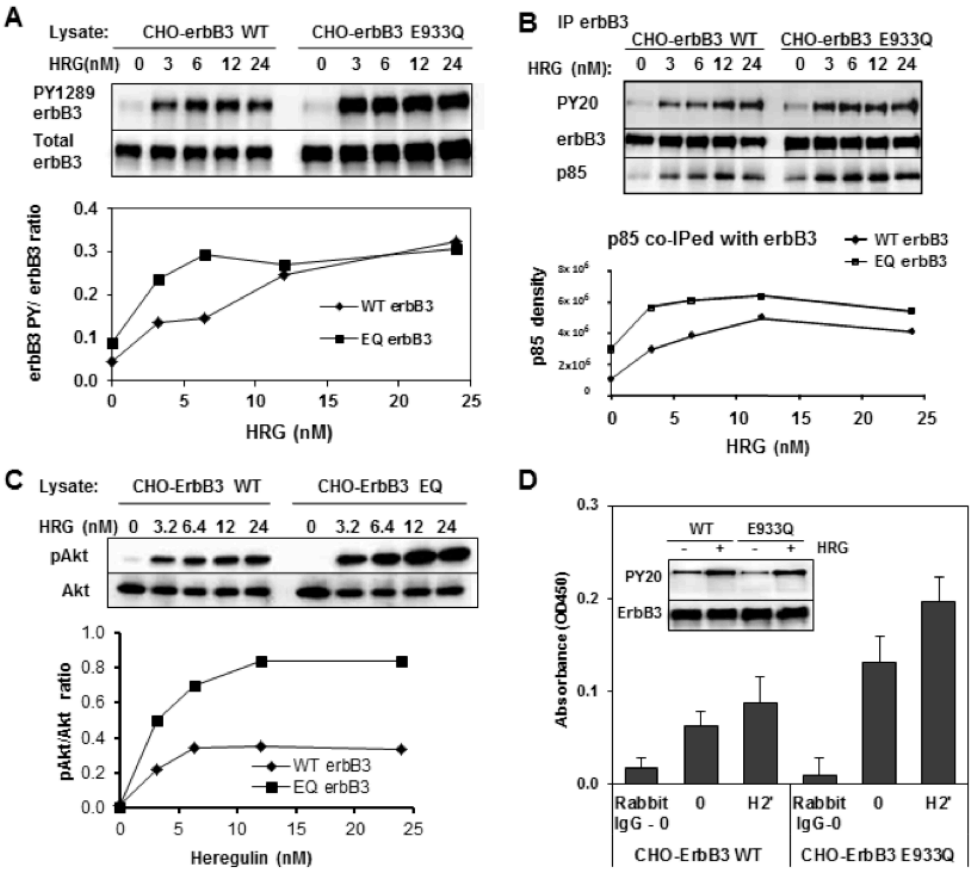

Figure 5.7: The gain of function mutant erbB3 E933Q demonstrates increased sensitivity to ligand and increased kinase activity when expressed in CHO cells. (A) CHO erbB3^{E933Q}-mCitrine cells have high levels of erbB3 phosphorylation even at the lowest dose of HRG (3 nM). (B) Immunoprecipitated erbB3 E933Q co-immunoprecipitates more p85, a PI3kinase subunit, in the presence of ligand. (C) Increased levels of phosphorylated Akt that lead downstream of PI3kinase indicates an overall upregulation of the erbB3 signaling network in CHO erbB3^{E933Q}-mCitrine cells. (D) Immunoprecipitated erbB3^{E933Q} has higher basal and ligand-dependent kinase activity in the in vitro kinase assay compared to erbB3^{wt}.

phosphotidylinositol-3-kinase (PI3K). Cells expressing erbB3^{WT} or erbB3^{E933Q} were stimulated over the same range of doses, followed by immunoprecipitation of erbB3 from cell lysates and western blotting to measure co-precipitated p85 regulatory subunits of PI3K. The amount of p85 recovered in ErbB3 immune complexes paralleled the degree of phosphorylation of ErbB3 in the CHO transfectants, with more p85 recovery at low ligand doses for ErbB3^{E933Q} than for ErbB3^{WT} (Figure 5.7B). Cells expressing ErbB3^{E933Q} also had a more robust activation of the downstream serine/threonine kinase, AKT, as measured by phosphorylation of AKT S473 (Figure 5.7C). The phosphoAKT/total AKT ratio was higher at times after HRG stimulation of CHO cells expressing ErbB3^{E933Q}, compared to cells expressing the wildtype receptor. Figure 5.7D reports results of *in vitro* kinase assays for erbB3^{WT} and erbB3^{E933Q} immune complexes, isolated independently from the cell lysates of the two transfected CHO cell lines after 2 min challenge with 12nM HRG. Greater activity is shown for mutant erbB3, despite modest differences in phosphorylation at this dose. Together, these results indicate that the E933Q substitution is an example of a class of gain-of-function mutations (Jaiswal et al, 2013) for which the mathematical model might provide valuable insight.

As a prelude to simulations, we applied our existing experimental methods to determine if the E933Q mutation might alter basic parameters in the model such as the diffusion rate, homodimer lifetime or dephosphorylation kinetics. Results of these experiments are shown in Figure 5.8A-D. Liganded erbB3 homodimers are slow <0.05 for all pairs composed of either WT or E933Q forms (Figure 5.7A). However, we observed slightly faster monomer diffusion coefficients for erbB3^{E933A}, whether co-

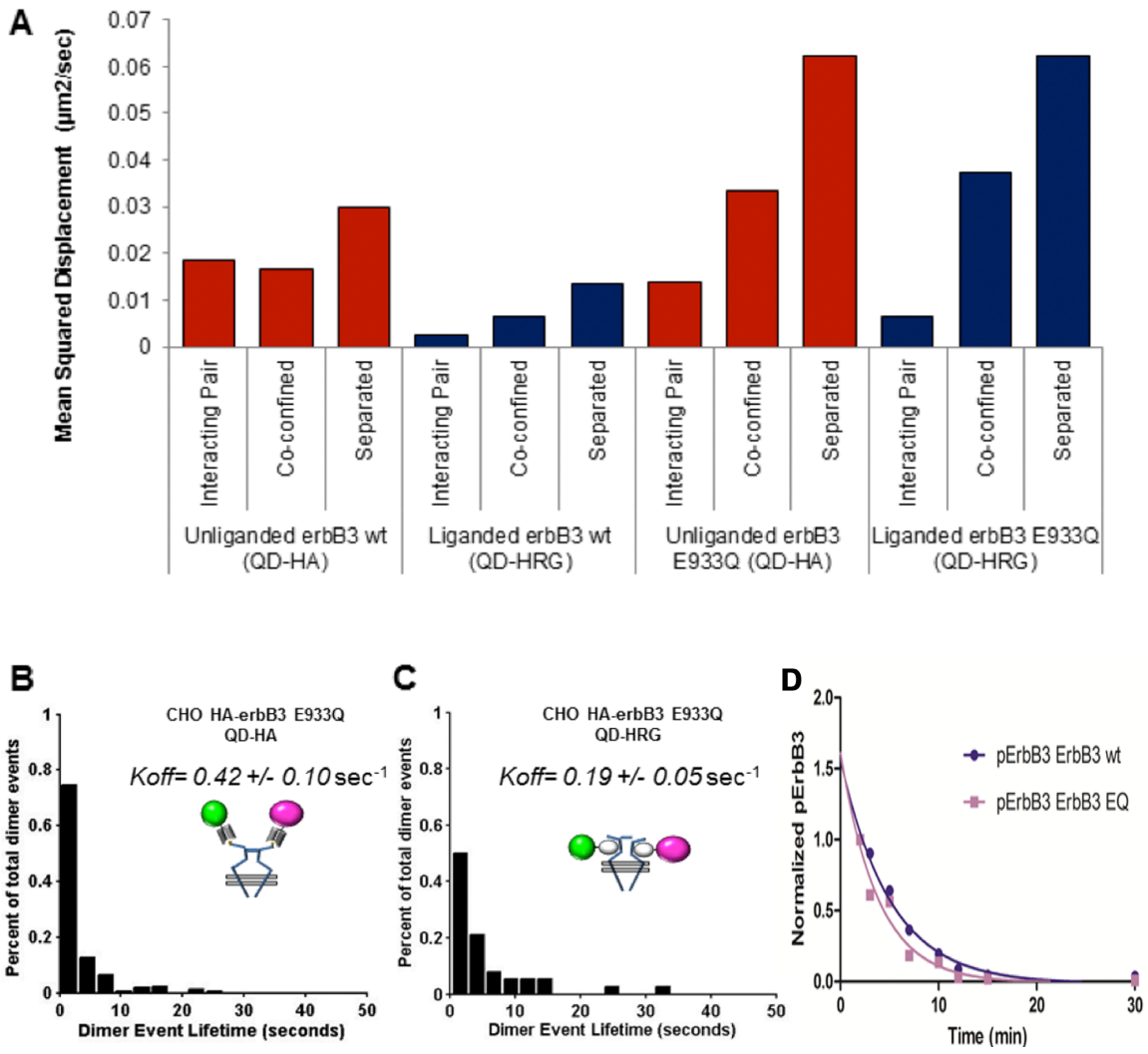

Figure 5.8: Single particle tracking of HA-erbB3^{E933Q}. (A) Diffusion by state comparing HA-erbB3 E933Q to HA-erbB3 wt. E933Q shows faster diffusion when separated or co-confined, but both unliganded and liganded interacting pairs diffuse slower than interacting HA-erbB3 wt pairs. (B) Unliganded dimer pairs are short lived as shown by the histogram of dimer lifetimes. Unliganded receptor pairs have an off rate of 0.42 sec-1 that is comparable to the wild type value. (C) Liganded HA-erbB3 E933Q receptor pairs are longer lived as seen with the wildtype receptor and have an off rate of 0.19 sec-1 that is equivalent to the wild type value. (D) Normalized phosphorylation levels for erbB3^{wt} versus erbB3^{E933Q} were plotted over time. As in Figure 5.4B, phosphorylation levels of both receptors were set to 1 for the two minute time point (maximum HRG stimulation). Points were fit to a one-phase exponential decay curve to determine the dephosphorylation half-life.

confined in domains or widely separated (Figure 5.7A). The off-rate of 0.41/sec for unliganded erbB3^{E933Q} homodimers (Figure 5.7B) is comparable to previously measured values for erbB3^{WT} (Steinkamp et al, 2014), as is the 2-fold slower off-rate of 0.19/sec for liganded pairs (Figure 5.7C). The plot in Figure 5.8D (derived from data shown in Supplemental Figure 5.4) shows that dephosphorylation kinetics for WT and E933Q forms of erbB3 are the same.

5.4 DISCUSSION

Here we have presented a combination of experimental and quantitative methods to explore the relationship between erbB2 and erbB3, as well as set a framework for evaluation of the growing list of erbB3 mutations. Starting from observations in the experimental data about receptor dynamics and membrane landscape, we were able to develop a new algorithm to deduce confinement zone shapes to be used in conjunction with a spatial stochastic model. Our spatial stochastic model was then able to not only reproduce the receptor dynamics observed in experiments, but also give further insight into the complex interplay between erbB2 and erbB3.

Previous work in both the experimental and modeling fields have given great insights to the erbB receptor family. Until recently it was believed that the kinase domain of erbB3 was completely dead. Shi et al. (2010) reported weak kinase activity for erbB3 cytoplasmic tails tethered to liposomes. In conjunction with evaluating erbB2/3 interactions by single molecule methods, we made the novel observation that erbB3 kinase activity was significantly upregulated in an erbB2-dependent manner and linked to its phosphorylation state (Steinkamp et al 2014). Although the putative phosphorylation site that activates erbB3 has not yet been identified, the erbB2-mediated activation of

erbB3 is a major assumption in our mathematical model. We use spatial stochastic modeling to extend that erbB2/3 analysis one step further, quantifying the relative contributions of erbB2 and erbB3 based upon their dependency upon asymmetric orientation of their kinase domains and their relative phosphorylation states. Our model incorporates the relative dimer lifetimes and diffusion rates for each species, showing that the more transient nature of ErbB2-3 heterodimers is a critical factor in the signaling process.

The results of our model indicate that erbB2 is critical to erbB3 activation. Dimer lifetime was also highlighted as an important factor in phosphorylation. ErbB3 homodimers are extremely stable causing the homodimer to become a hindrance for erbB3 receptor phosphorylation. Simulations showed that erbB3 receptors quickly formed homodimers with one another before having the chance to interact with erbB2. This caused a lack of phosphorylation capacity between erbB3 receptors, since there was no activation by erbB2. The erbB3 receptors that were able to interact with erbB2 receptors tended to interact again with other erbB2 receptors because the other erbB3 receptors were tied up in unproductive homodimers.

Our simulation results are in good agreement with the experimental data, giving confidence to broaden the model beyond wt ErbB3 interactions. As a preliminary exploration, we implemented a gain of function mutation that caused an increase in activity of the C-lobe of the mutant erbB3 receptor. Varying the intensity the mutation has on the phosphorylation rate gave insight into how this mutation would impact erbB2 and erbB3 activation and reliance on one another. Our discovery of the E933Q erbB3 gain of function mutation gave us experimental data to relate our mutation simulation to

real experimental data. Extending our model to look at the impact of erbB3 mutations is a characteristic of the experimental-modeling handshake. We can now use our model to make predications about how different mutations will affect the dynamics and kinetics between erbB2 and erbB3 receptors, giving experimentalists a good idea of which mutations may be advantageous to investigate further. Interesting mutation to study in future work, related to receptor activation, could cause a change in receptor conformation stability, longer or short dimer lifetimes, and/or different kinase domain activation levels.

5.5 MATERIALS & METHODS

5.5.1 Experimental

5.5.1.1 Determination of the dephosphorylation rates for erbB2 and erbB3

CHO cells stably transfected with a pcDNA3.1- HA-erbB2 construct were transiently transfected with a pcDNA3.1-erbB3-GFP construct and seeded into six well dishes. Two days after transfection, cells were serum starved for up to four hours, and then stimulated for two minutes at 37^{o}C with 12 nM (75 ng/ml) heregulin-β (US Biologicals). Cells were then treated with 10 μM lapatinib (Santa Cruz) to inhibit erbB2 kinase activity and harvested 1-30 minutes after HRG activation. Controls were harvested at time 0 (no HRG stimulatiom), time 2 min. (HRG stimulation –lapatinib) and time 30 (HRG stimulation – lapatinib). Cells were harvested in lysis buffer and total protein levels were determined by BCA assay. 20 ug of total protein/sample was run on an acrylamide gel and transferred to nitrocellulose membranes using the iBlot system (Invitrogen). Phosphorylated erbB3 was detected using a PY1289-specific rabbit polyclonal antibody (Cell Signaling) and total erbB3 protein was detected using an anti-erbB3 monoclonal rabbit antibody (Cell Signaling). Phosphorylated erbB2 was detected using a PY1248-

specific rabbit polyclonal antibody (Cell Signaling) and total erbB2 was detected using an anti-erbB2 monoclonal mouse antibody (Thermo Scientific). Secondary anti-rabbit and anti-mouse-HRP were from Santa Cruz Biotech. Blots were revealed using the SuperSignal West Pico Chemiluminescent Substrate (Pierce) and detected on a ChemiDoc Gel Imager (BioRad). Bands were quantified using ImageLab quantification software (BioRad). Data was plotted and fit to a one phase exponential decay curve using GraphPad Prism5. The dephosphorylation half-life was calculated from this curve.

5.5.1.2 Immunoprecipitation

CHO cells expressing ErbB constructs were serum starved for up to 4 hours. CHO ErbB3^{wt}-mCitrine were stimulated with 12 nM HRG for 2 minutes. All cells were washed with ice cold PBS on ice and lysed with cold NP-40 lysis buffer (Yang et al, 2007). Protein concentrations in cleared lysates were measured by BCA assay (Pierce, Rockford, IL). Supernatants were precleared with Protein A-beads (Amersham GE Healthcare, Chicago, IL) and normal rabbit IgG (Cell Signaling) followed by overnight incubation at 4°C with primary antibodies against erbB2 (RB9040, Neomarkers) or erbB3 (sc285, Santa Cruz). Proteins in washed immune complexes were either resuspended in reaction buffer for the in vitro kinase assay or denatured and separated by SDS-PAGE, transferred to nitrocellulose membranes, and probed with anti-PY20 (Sanat Cruz) to detect all phosphorylated proteins in the IP or with primary antibodies against erbB2 or erbB3 and HRP-conjugated secondary antibodies. Blots were revealed and detected as above.

5.5.1.3 Sequencing of erbB3 from SKBR3 cells

mRNA was extracted from SKBR3 cells using the RNeasy Mini Kit (QIAGEN, Chartsworth, CA). Pairs of primers were designed to span the ErbB3 tyrosine kinase domain: 5'-CTC TGG ACC CCA GTG AGA AG-3' and 5'-GGG AGT ACA AAT TGC CAA GG-3'; 5'-GGT CAG CCA CAC CAA AAT CT-3' and 5'- CAG ATA CCG TGG TGG GTC TC-3'. After amplification using a QIAGEN One-step RT-PCR kit, PCR products were separated by agarose gel electrophoresis, extracted with the QIAquick gel extraction kit, and sequenced using the above primer sets.

5.5.1.4 In vitro kinase assay

The in vitro kinase assay was performed as detailed in (Steinkamp et al, 2014). Briefly, CHO cells expressing either erbB3^{wt}-mCitrine or ErbB3^{E933Q}-mCitrine were serum starved then stimulated for 2 minutes with 12 nM HRG-β for maximal erbB3 activation. Lysates were immunoprecipitated with the anti-ErbB3 polyclonal rabbit antibody, sc-285 (Santa Cruz Biotech).

5.5.1.5 Single particle tracking

The E933Q mutation was introduced into an expression vector pcDNA3.1-HA-erbB3 and stably transfected into CHO parental cells. CHO HA-erbB3E933Q cells were sorted for high expression after labeling with an anti-HA-Alexa488 antibody (Cell Signaling). Single particle tracking experiments were performed as in (Steinkamp et al, 2014). Briefly, CHO HA-erbB3E933Q cells were plated in 8-well chambers (LabTek) the day before imaging. Heregulin was biotinylated using NHS-ester chemistry with Biotin-XX-SE (Invitrogen) and purified from free biotin on a G25 column by gravity flow. Biotinylated HRG and biotinylated anti-HA Fab were conjugated to streptavidin QDots

(Invitrogen) QD655 and QD585, in a 1:1 molar ratio in PBS + 0.1% BSA by rotating for 1 hour or more at 4oC. Cells were serum starved for 1 hour and then moved into imaging medium, Tyrodes buffer supplemented with 1% BSA and 20 mM glucose. Cells were labeled with 100 pM QD-anti-HA Fab for 15 min. or with 20 pM QD-HRG for 5 min before washing and imaging at 20 frames/s on an Olympus IX71 inverted microscope with a 60× 1.2 N.A. water objective. An objective heater (Bioptechs, Butler, PA) was used to stabilize chamber temperature at 34-35 °C. A mercury lamp with a 436/10 nm BP excitation filter provided wide-field excitation. Emission was collected by an electron multiplying CCD camera (Andor iXon 887) using a DuoView image splitter (Optical Insights) to image both the QD585 (585/20 BP) and QD655 (655/40 BP) probes. Image processing was performed using Matlab (The MathWorks, Inc., Natick, MA) functions in conjunction with the image processing software DIPImage (Delft Univ. of Technology). Single molecule localization, trajectory elongation, and 2-channel image registration were performed as previously described (Low-Nam et al, 2011; Smith et al, 2010; Steinkamp et al, 2014).

5.5.1.6 State determination of receptor pairs

Receptor interactions were identified using a three-state Hidden Markov Model (HHM) (Low-Nam et al, 2011). The separation between two color QD pairs (one QD585 and one QD 655) is determined and pairs that are less than 1 micron apart are considered candidate pairs. Distribution of the displacements between the QDs is modeled by a zero mean Gaussian distribution in each (x,y) dimension using σ_{dimer} and σ_{domain} respectively. The value σ_{dimer} for erbB3 was estimated by combining information from EGFR crystal structure measurements, homology modeling, and the size of QDs (Steinkamp et al,

2014). For the Separated state in the model, the probability density is calculated as a function of the observed distance in the previous frame and a characteristic diffusion constant. Rate constants are determined by maximizing the likelihood over all interactions of two QDs for a specific condition. Standard errors for parameters are calculated as $(H_{i,i-1})$-0.5 where H is the Hessian matrix of the negative log-likelihood and i denotes one of the estimated rate constants. The Viterbi algorithm (Forney, 1973) is used to identify the most likely state within individual QD interactions. Diffusion by state was calculated based on the mean square displacement of all tracks per condition.

5.5.2 Modeling

5.5.2.1 Modeling Biological Assumptions

Our model assumes that erbB3 dimerization occurs through conventional interactions between dimerization arms of the extracellular domains (Yarden & Sliwkowski, 2001). This premise is supported by evidence that blockade of the erbB2 dimerization arm with 2C4 antibodies ablates its transactivation of erbB3 (Steinkamp et al, 2014; Zhang et al, 2009), as well as homology models of erbB3 based upon the structure of EGFR (not shown). We do not formally consider the possibility that erbB3 may engage in homo-interactions through other interfaces (Zhang et al, 2012), due to the lack of available kinetic parameters. We also do not consider the intriguing possibility for higher order oligomers (Kani et al, 2005; Kozer et al, 2013), since the sparse labeling of SPT renders it highly unlikely that such events can be captured and measured using our available technologies.

5.5.2.2 Non-spatial modeling of dephosphorylation kinetics

We used BioNetGen to create a non-spatial version of our model to investigate the dephosphorylation kinetics. Dephosphorylation is a space independent reaction, therefore we used a non-spatial model to allow us to look over a longer period of simulation time to see the impact. The parameters and reaction network are the same as the parameters used for the spatial stochastic model.

5.5.2.3 Spatial stochastic model for homo and heterodimerization.

Domain Reconstruction Algorithm. To score each point in each trajectory from a single file and channel, the length of the jump is compared for varying step sizes. Compiling and comparing these scores against one another reveals a bimodal distribution in the scores. The local minimum of the modes is used as a cutoff score. This score separates the trajectory points in to confined points and free points. This step takes the dynamic SPT data and creates a subset of static data. The slow points are considered to be representative of confined areas on the membrane. The slow points are then clustered to determine confinement areas or domains. A characteristic length study is first done to determine the characteristic length that best describes the distance between points in a cluster. This length is varied from 0nm to 1000nm. Over this range the average perimeter of the clusters is computed. The perimeter reaches a local minimum over this range. The characteristic length where this minimum is reached is used as the characteristic length. The characteristic length is then used to create the clusters. Each cluster is then used to create a contour that will define the final confinement area. The contour is created by "inflating" the cluster. The outside points of the cluster are extended outward by ½ the characteristic length. This allows some give in the confinement area to consider that these

points could have bounced off a structure and would not have sat directly next to the confining structure. The extended points are then connected to create a contour. This is the final domain structure.

Reaction Network Generation. Our reaction network definition begins with the interaction model proposed by Steinkamp et al. (2014). In this model, ErbB2 receptors dimerize with ErbB3, allowing ErbB2 to activate (Phosphorylate) ErbB3. These dimer interactions appear to be transient, and quickly fall apart. The activated ErbB3 receptor is then free to homo-dimerize with another ErbB3. The activated state of the ErbB3 allows it to illicit kinase activity from the unphosphorylated ErbB3 causing activation and therefore phosphorylation. We extend this model slightly by implementing the asymmetric phosphorylation model used in our previous ErbB1 activation paper (Pryor et al, 2013).

Receptor Diffusion and Reaction Kinetics. Receptors diffuse according to Brownian motion through a combination of specific diffusion coefficients calculated from SPT experiments and normally distributed random variables. Diffusion is interrupted when a membrane domain is encountered in the receptors path. To account for this obstacle two tolls are considered, entrance into the domain and exit from the domain. The entrance and exit rates of the domains are converted to probabilities using the simulation time-step. When a receptor attempts to enter or exit a domain, the relevant probability is evaluated and the receptor is allowed to either leave the domain or the receptor is reflected back into the domain.

Receptor reactions are treated differently depending on their reaction order. 1st order reactions are treated similarly to the domain entrance/exit tolls. The reaction rate is

converted to a probability using the time-step and evaluated. If the probability is met, the reaction occurs, if it is not met, no reaction occurs. Second order reactions are a bit more complicated. These reactions are diffusion limited, and therefore the spatial aspect of the model must be taken into consideration. The solution to this problem, proposed by Andrews and Bray (2004), is to use a binding radius and unbinding radius, if the reaction is reversible (Andrews & Bray, 2004). The binding radius is a function of simulation time-step, receptor diffusion coefficients, and receptor on rates. At the end of a receptors diffusion step, the area around the receptor bounded by the binding radius is scanned for other receptors. If an available receptor is found within the binding radius, a reaction occurs. There is no probability associated with second order reactions; likelihood of reaction is taken into consideration through the binding radius. If a reverse reaction is possible, an unbinding radius is implemented. The length of the unbinding radius is picked such that an unrealistic number of repeated interactions are minimized. Andrews and Bray suggest the binding to unbinding radius ratio be 20% as a starting point.

5.6 NOTES

This work is currently being prepared for submission for peer review.

5.7 ACKNOWLEDGEMENTS

This work was supported by NIH-CA119232 (BSW), NSF MCB-0845062 (DSL) and P50GM085273 (BSW), NIH grants R01 GM104973 (to JSE and AMH) K25 CA131558 (AMH). MMP was supported by the U.S. Department of Energy through the LANL/LDRD Program for this work. Use of the UNM Cancer Center Microscopy Facility, and NIH support for instruments and staff support, is gratefully acknowledged.

CHAPTER 6: DISCUSSION

6.1 SUMMARY

The focus of this dissertation was to integrate data from live cell imaging into a spatial stochastic model to be used to study the kinetics and dynamics of the ErbB family of tyrosine kinase receptors. We started by building a spatial stochastic model using an EM image as our simulation space and parameters derived from SPT tracking of ErbB1 receptors. We were able to validate our model by replicating the SPT data. We used that model to investigate how membrane landscape and receptor interactions impact receptor state in ErbB1 receptors. We then moved on to analyze SPT even further to uncover anomalous diffusion characteristics and how the membrane landscape related to those dynamics. Taking the SPT analysis a step further, we developed an algorithm to reconstruct confinement zones directly from the SPT data. Finally, we used the reconstructed confinement zones from ErbB2 and ErbB3 tracking data to create a simulation space for our stochastic model, while extending the spatial stochastic model to a heterogeneous population to study ErbB2 and ErbB3 interaction dynamics.

6.2 SIGNIFICANCE

Our spatial stochastic model agrees well with existing experimental data, and enabled us to look deeper into the mechanisms behind erbB receptor signaling beyond current experimental capabilities. Through each of our modeling studies we were able to highlight the importance of the cellular membrane landscape and how it impacts receptor activation for signaling. We also characterized the distribution of confinement zones on the membrane landscape and the size and shape of the confining areas present, gaining

important insight as to what to type of structures could contribute to the confinement zones.

6.2.1 Spatial Stochastic Model

Our spatial stochastic model directly used SPT experimental data without the need to do any parameter fitting, giving our model a very strong connection to the biology it was built to simulate. Our model is extremely useful for studying how different protein dynamics, typically studied without the ability to consider the impact of one another, come together. This view allows us to get a more comprehensive look at how all the factors impact receptor signal initiation. Our model nicely lends itself to be built upon further in the future to either study erbB receptors more in-depth or branch out to a different transmembrane protein system.

6.2.2 ErbB1 Receptor State Dependence on Membrane Landscape

Through our ErbB1 study, we showed that spatial aspects of the cellular membrane are a critical element to take into consideration when investigating membrane receptor dynamics. These findings are very important for the membrane and erbB receptor research communities, as this is a source of debate. We show that confinement zones on the membrane create small zones of high-density receptors, causing an increase in receptor interactions. These interactions lead to higher dimer phosphorylation and activation, impacting the extent of down stream signaling in cells.

6.2.3 Anomalous Diffusion, Membrane Landscape, and Domain Reconstruction

Building off the conclusion of the ErbB1 study, we decided to delve deeper into the cellular membrane landscape. Analysis of SPT tracking data of FcεRI receptors revealed non-Brownian motion features. We found that these diffusion characteristics could be

contributed to a membrane landscape consisting of attractive domains, but not for a landscape made up of corrals. Insight into the size and distribution of these confinement zones is useful for future work determining the actual structure or structures that cause these diffusion anomalies in cell membranes.

6.2.4 ErbB2 – ErbB3 Interactions and ErbB3 Gain of Function Mutation

In our ErbB2/3 study, we employed a combination of experimental and modeling methods. We were first able to use our SPT analysis to show that erbB2 and erbB3 receptors have significantly different diffusion characteristics. Building from the previous ErbB3 kinase activity study by Steinkamp et al. (2014), we showed that erbB3 can sustain activation after being activated by erbB2, indicating erbB3 receptors have kinase activity, and were able to relate erbB2 phosphorylation to erbB3 phosphorylation to calculate a phosphorylation rate multiplier for erbB3. These results alone have an enormous impact on the current understanding of erbB3 activation and signaling. Applying this information to our spatial stochastic model, we were able to confirm the dependence of erbB3 activation on erB2 interactions. We also concluded that the stability of erbB3 homodimers is actually an obstruction to erbB3 activation.

Agreement between the erbB2/3 experimental results and our model provide a high level of confidence in our model predictions. We decided to extend the model beyond wt interactions, and investigate how a gain of function mutation in the kinase tail of erbB3 would alter erbB2 and erbB3 phosphorylation. We found that, while erbB2 phosphorylation was not drastically impacted, erbB3 phosphorylation showed a strong dependence on the kinase domain activity. We were also able quantify how strong the gain of function would have to be to eliminate erbB3 dependence on erbB2 activation.

We discovered an E933Q amino acid swap gain of function mutation in erbB3, which allowed us to study the kinetics and dynamics of the mutated receptor. We show that this mutation does not impact erbB3 homodimer lifetimes; we therefore conclude that this mutation may cause an uptick in kinase activity, allowing us to compare out experimental results with our model predictions. The ability of the model to predict outcomes of mutations provides great insight in to what types of mutations could be advantageous to study more in-depth.

6.3 FUTURE DIRECTIONS

Our spatial stochastic model can continue to be used and extended for deeper research into erbB receptor dynamics, or can be modified to study an entirely different membrane protein system. The Domain Reconstruction Algorithm can also be used to study and characterize other proteins whose diffusion can be tracked individually, in real time. Here we discuss a few possibilities for the future use of our model.

6.3.1 Current Path Forward: ErbB3 Mutation Study

In our erbB2/3 study, we used our model to investigate a mutation in the kinase domain of erbB3. We propose, and are currently in the process of, using the model to look into other gain of function mutations for erbB3. Jaiswal et al. (2013) compiled a list of known erb3 mutation, we add our E933Q gain of function mutation to this list. We can then use our model to investigate mutations that have yet to be discovered. The model will give insight into the type of gain of function mutation that will have the greatest impact on erbB3 activation. These mutations can then be replicated in real erbB3 receptors and studied for different therapeutic uses.

Examples of possible high impact mutations that would change receptor activation include alterations in receptor conformation flux, dimer lifetimes, and/or receptor kinase activity. Altering the stability of unliganded erbB3 receptors through receptor conformation decreases erbB3's dependence on ligand. A more stable unliganded receptor would allow higher dimerization rates with other erbB2 and erbB3 receptors in the absence of ligand. Changing the lifetime of erbB3 homo and hetero-dimers would impact the ability of erbB3 receptors to become phosphorylated. In our wt erbB2-3 model, we showed that the long dimer lifetimes of erbB3 dimers negatively impacts erbB3's ability to become activated. Varying the dimer lifetimes of hetero and homo-dimers would allow for a balance to be struck to either up-regulate or down-regulate receptor phosphorylation. Finally, a mutation in the kinase domain, similar to the E933Q mutation we found, would directly alter receptor activity. We believe our E933Q mutation causes an uptick in activity in the C-lob of the erbB3 receptor, making it a stronger activator in receptor phosphorylation. The mutation we propose here to study would be located in the N-lobe of the kinase domain, making the erbB3 receptor a strong receiver.

6.3.2 For Future Investigators

We originally built the Domain Reconstruction Algorithm with characterizing erbB2 and erbB3 domains in mind, however it is easy to see that this algorithm can be used for any type of trajectory data. The analysis can be used to characterize confinement zones for other transmembrane proteins, such as MET, FcεRI, or Toll-like Receptors. A useful project for the DRA would be to characterize the confinement zone characteristics for a specific protein for various cell lines. This would give insight into how much the cellular

membrane landscape varies between cells, and also how the membrane landscapes vary between "normal" cells and cancerous cells. It can also be extended to characterize confinement of other types of proteins that can be tracked individually. This algorithm can even be extended beyond biological systems, to characterize diffusion characteristics of particles or diffusion-limited chemical species.

Beyond using the spatial stochastic model to investigate different erbB3 mutation scenarios, the model can quickly be altered to investigate the interaction dynamics of another diffusion dependent protein family that forms dimers. No change to the actual code would be necessary, only the input parameters dictating diffusion and receptor reactions would need to be updated to match the protein system.

To extend the model further, protein families that form higher order oligomers could be studied by altering the dimerization rule in the model. This would be an interesting next step for the model, as oligomerization of erbB receptors has been a popular research focus (Kani et al, 2005; Kozer et al, 2013). This rule change for the model would also allow the model to be used to study other proteins known to form oligomers, such as FcεRI.

A more drastic change to the model would be to add a 3D component to represent the cytoplasm. This addition would allow for investigation deeper into the proteins involved in the signaling cascade initiated through membrane receptor activation. Members of our group previously explored this type of model using a lattice-based spatial stochastic model (Costa et al, 2009b). Since this work was published, experimental methods have improved, giving more precise parameters for model building.

6.4 PERSPECTIVES

During this dissertation we worked hard to directly integrate experimental data with our modeling methods. In each study we used experimental data to create our model, then extended the model beyond current experimental capabilities. This extension shows the usefulness of modeling for scientific research. Models give us deeper insight and greater flexibility when studying the intricate relationships and mechanisms of physical processes. Integration of experiments and modeling is a crucial step for the future of scientific research. Creating a continuous cycle of experiments and modeling will allow us to expedite scientific discoveries by using models to predict interesting and relevant mechanisms to study leading to novel experiment development.

APPENDICES

APPENDIX A: Simulation Space Program Scripts

SIMULATION IMAGE IMPORTER

```
%% EM Image Importer m-file
% Final Version by Meghan McCabe Pryor April 16th, 2014
%% Fresh Start
clear all; close all; clc
%% Input information
% Name of save file
datafilename='EMFileName';
% Name of image to be used for particle positions
photoname='5-10957Large_ScaleBar.jpg';
% Unit conversion for scale bar on image
scalebar=0.1; % Enter in units of um
%% Check File Name Existance
% To avoid overwritting past data files
if exist(strcat(datafilename,'.mat'),'file') == 2
   % Check before overwritting savefilename
   choice1 = questdlg('Data Filename already used, are you sure you want to overwrite
the file?', ...
      'Save Name Check', ...
      'Yes','No','No');
   switch choice1
      case 'Yes'

      case 'No'
         disp('Please rename the save file name and re-start the simulation.')
         return
   end
end
%% Data Import
choice2 = questdlg('Do you want to include domains?', ...
   'Domain Check', ...
   'Yes','No','No');

switch choice2
   case 'Yes'
      %%
      % Import image and show
      EM = imread(photoname);
      imshow(EM)
      hold on
      %%
```

```
% How many domains?
prompt = {'How many domains are there?'};
dlg_title = '# of Domains';
num_lines = 1;
answer = inputdlg(prompt,dlg_title,num_lines);
numdomain=str2double(answer{1});
%%
% Measure scale bar distance
h1=msgbox('Scale Bar Length (2 Clicks)');
uiwait(h1)
[d1 d2] = ginput(2);
%%
% Measure x axis and y axis
h2=msgbox('x-axis Length (2 Clicks)');
uiwait(h2)
[px nay] = ginput(2);
h3=msgbox('y-axis Length (2 Clicks)');
uiwait(h3)
[nax py] = ginput(2);
%%
% Measure x axis and y axis
r=zeros(1,3);
for i = 1:numdomain
   h5=msgbox(sprintf('x-axis Length of Domain %.0f (2 Clicks)',i));
   uiwait(h5)
   [idpx nay] = ginput(2);
   dpx(i,:)=idpx';
   h6=msgbox(sprintf('y-axis Length of Domain %.0f (2 Clicks)',i));
   uiwait(h6)
   [nax idpy] = ginput(2);
   dpy(i,:)=idpy';
   dxmin=idpx(1);
   dymin=idpy(2);
   dxmax=idpx(2);
   dymax=idpy(1);
   plot([dxmin dxmax dxmax dxmin dxmin], [dymax dymax dymin dymin
dymax],'b')
   %%
   % Measure distance ***Press Return when finished clicking***
   h4=msgbox(sprintf('Particle Placement in Domain %.0f (Right Click when
finished)',i));
   uiwait(h4)
   iXp=[];
   iYp=[];
   %key = '';
   key = 1;
```

```
    while key == 1%strcmp(key,'')
        [Xp Yp key] = ginput(1);
        iXp=[iXp Xp];
        iYp=[iYp Yp];
        plot(iXp,iYp,'r.')
        %key = get(gcf,'CurrentKey');
    end
    %key=set(gcf,'CurrentKey','');
    key = 1;
    [m n] = size(r);
    r(m+1:m+length(iXp),:)=[iXp' iYp' i*ones(length(iXp),1)];
end
r(1,:)=[];
%%
% Measure distance ***Press Return when finished clicking***
h4=msgbox(sprintf('Free Particle Placement (Right Click when finished)'));
uiwait(h4)
key=1;
iXpfree=[];
iYpfree=[];
while key == 1; %strcmp(key,'')
    [Xpfree Ypfree key] = ginput(1);
    iXpfree=[iXpfree Xpfree];
    iYpfree=[iYpfree Ypfree];
    plot(iXpfree,iYpfree,'r.')
    %key = get(gcf,'CurrentKey');
end
key=1;
%%
% Scale Units
unit=abs(d1(2)-d1(1))/scalebar; % Converts bewteen pixels and um [=] # pixels/um
xlimmax=px(2)/unit; % x limit in um
ylimmax=py(1)/unit; % y limit in um
xlimmin=px(1)/unit; % x limit in um
ylimmin=py(2)/unit; % y limit in um
% Domain limits
dxmin=dpx(:,1)/unit;
dymin=dpy(:,2)/unit;
dxmax=dpx(:,2)/unit;
dymax=dpy(:,1)/unit;
%%
% Add free data points to domain data points
rcombine(:,1) = [r(:,1)' iXpfree]';
rcombine(:,2) = [r(:,2)' iYpfree]';
```

```
r(length(r(:,1))+1:length(rcombine),:)=[rcombine(length(r(:,1))+1:length(rcombine),:)
zeros(length(rcombine)-length(r(:,1)),1)];
    %%
    % Scale collected data
    iX=rcombine(:,1)./unit;
    iY=rcombine(:,2)./unit;
    close all
    %%
    % Plot initial data compared to image
    figure
    subplot(1,2,1);
    plot(iX,iY,'.', [dxmin dxmax dxmax dxmin dxmin]', [dymax dymax dymin dymin
dymax]')
    xlim([xlimmin xlimmax])
    ylim([ylimmin ylimmax])
    set(subplot(1,2,1),'YDir','reverse')
    subplot(1,2,2)
    imshow(EM)
    hold on
    plot(rcombine(:,1),rcombine(:,2),'.')
    hold off
  case 'No'
    %%
    % Import image and show
    EM = imread(photoname);
    imshow(EM)
    %%
    % Measure scale bar distance
    h1=msgbox('Scale Bar Length (2 Clicks)');
    uiwait(h1)
    [d1 d2] = ginput(2);
    %%
    % Measure x axis and y axis
    h2=msgbox('x-axis Length (2 Clicks)');
    uiwait(h2)
    [px nay] = ginput(2);
    h3=msgbox('y-axis Length (2 Clicks)');
    uiwait(h3)
    [nax py] = ginput(2);
    %%
    % Measure distance ***Press Return when finished clicking***
    h4=msgbox('Particle Placement (Hit Enter when finished)');
    uiwait(h4)
    [iXp iYp] = ginput;
    %%
```

```
        % Scale Units
        unit=abs(d1(2)-d1(1))/scalebar; % Converts bewteen pixels and um [=] # pixels/um
        xlimmax=px(2)/unit; % x limit in um
        ylimmin=py(2)/unit; % y limit in um
        xlimmin=px(1)/unit; % x limit in um
        ylimmax=py(1)/unit; % y limit in um
        %%
        % Scale collected data
        iX=iXp./unit;
        iY=iYp./unit;
        close all
        %%
        % Plot initial data compared to image
        figure
        subplot(1,2,1);
        plot(iX,iY,'.')
        xlim([xlimmin xlimmax])
        ylim([ylimmin ylimmax])
        set(subplot(1,2,1),'YDir','reverse')
        subplot(1,2,2)
        imshow(EM)
        hold on
        plot(iXp,iYp,'.')
        hold off
end
NP=length(iX);
%% Save Imported Data
save(datafilename)
```

DOMAIN RECONSTRUCTION ALGORITHM

The scripts presented in this appendix are still a work in progress. The script posted here are up-to-date as of the writing of this document.

Domains_V006.m

```
% Labels points based on step size distributions %
%
% V001 - can make histograms or not, labels points based on a percentage
% cutoff
%
% V002 - file and folder name separate, can use rankings
%
% V003 - added combination scoring
%
% V003a - cleaning up
```

```
%       - multi zoom saves
%
% V004 - multi-file processing
%      - analyze a set of files defined by a string token
%      - output two-channel results in a single proecessed data file
%      - build a cumulative step distribution from all files (by channel)
%      - for each file and channel, do the scoring and short points
%      - cluster the short points
%      - put out detailed clustering data for one length (MyLength)
%
% V005
%     - add domain diameter to geometric measures of domains
%     - length study option for short points
%     - two different MyLength values for the two channels (V005a)
%
% V006
%    TODO:
%     - area calculation for all (short and other) points -- done
%     - put out detailed clustering data for selected lengths
%     - use this to pick a common cutoff (one for each channel) - do this by hand?
%
%     Beyond this:
%     - look at overlap of trajectories and domains
%     - overlay plots with trajectories, domains from 1 or 2 channels
%     - estimate fraction of area covered by domains
%     - something else..?

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    *** EDIT THIS AS NEEDED ***                        %

% folder for source files
FolderPath    = 'CHO/CHO_HA-ErbB2_ErbB3_ErbB3/';
%FolderPath    = 'CHO/CHO_HA-ErbB2-ErbB3_ERBB2/';

% String pattern to help pick out a group of files %
%         make it blank (') to process all the files
SpecialString =  '';%'2013-3-29';%'2013-3-29-10-42';%'2012-12-19';%'';%

% extensions for source files (two channels)
Ext={'.HMMData_ch1_StDis2.mat','.HMMData_ch2_StDis2.mat'};


% length parameters used to define clusters and contours
%MyLength={0.0305,0.0570}; %{0.0570,0.0305}; %  %0.070;%0.035;
```

```
MyLength={0.1,0.1};

% for the length study option
% (substring to be found in the file names, set it to 'zumm'
% to turn off length study, set it equal to SpecialString to
% make it default
LS_SpecialString = 'zumm';%;'-3-29';

% folder for output files
% OutputFolder = 'Scratch/SampleRunsForPlots_ErbB32/';
OutputFolder = 'Scratch/Run005a/';

% extension+tag for processed files
OutputExt = {...
   sprintf('_DomRec_L%dnm.mat',floor(1000*MyLength{1})),
   sprintf('_DomRec_L%dnm.mat',floor(1000*MyLength{2}))};

% ** Processing / Analysis Parameters **

% step count values we expect in the input files
SelectStepVals=[1 2 5 10 20 50 100 200];

% Step Count Values (1,2,5,10,20,50,100,200)
% that will be used in building the score, for example:
% [3] means we only use the 5 step values
% [3 4 4 5] means 5 steps, 10 with double weight, and 20
%ScorePattern=[4 5 6];
ScorePattern=[1 2 3 3 4 4 5 5 5 6 6 7 7 8 8];% a combo score I liked

% cutoff used to define the short points (score below means short - max is 1)
CutFactor = 1.01;%0.65;

%   ** Optional Figures and Data Outputs **  %

MakePlot1001 = false; % cumulative step distribution plot
MakePlot404  = false; % individual histogram of scores (one per file)
MakePlot405  = false; % cumulative histogram of scores (one for the entire group)
MakePlot2001 = false; % length study plot

FILEOUTPUT   = false; % text file outputs of contours and points etc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%

% get a file list
DirOut = dir([FolderPath '*' SpecialString '*' Ext{1}]);
```

```
FileCount = size(DirOut,1);

FileTrajectoryCount = zeros(FileCount,2); % counts the points in each channel of each file
FilePointCount = zeros(FileCount,2); % counts the points in each channel of each file

ShortFileNames = cell(FileCount,1);


%% Global step size statistics %%

% Initialize the global step statistics object
% NOTE: make sure there is enough space

StepCounts=zeros(2,8); % keeps track of the number of values in the statistic

StepStats = cell(2,8);% holds the values of the step sizes



% best guesses for the size of the statistics

SSizes=[450000, 250000, 110000, 65000, 35000, 15000, 7000, 3000];
for ich=1:2,
   for isz=1:8,
      StepStats{ich,isz} = zeros(SSizes(isz),1);
   end;
end;

tic
% Collect metadata for the step statistics
for ifile=1:FileCount % initial loop through files (defined by the short name)

   % short file name from the dir() output
   %q=strsplit(DirOut1(ifile).name,'.');% works only in v2013 or later
   q=strread(DirOut(ifile).name,'%s','delimiter','.');% may be obsolete, use the above

   ShortFileNames{ifile} = q{1};

   fprintf('%s\n',ShortFileNames{ifile});

   for ich=1:2 % loop through channels

      File = [FolderPath ShortFileNames{ifile} Ext{ich}];
```

```
    if ~exist(File,'file'), % get trajectory, point and step info from the file
       fprintf('File %3d ch%d: %s not found\n',ifile, ich, File);
    else
       fprintf('ch%d: ',ich);
       load(File);

       FileTrajectoryCount(ifile,ich)=size(Trajectories,1);
       for TrNo = 1:FileTrajectoryCount(ifile,ich); %loop over trajectories and count the
points in them
          FilePointCount(ifile,ich) = FilePointCount(ifile,ich)+Trajectories{TrNo,2};
       end;

       fprintf('TrCt:%3d                                                    PtCt:%4d
',FileTrajectoryCount(ifile,ich),FilePointCount(ifile,ich));

       % error check: the pixel size should come with the data
       if ~exist('PixelSize','var'), fprintf('      Domains **error**      No PixelSize
found!\n'); return; end;

       L2ConversionFactor=PixelSize^2;

       % add the step values to the global statistics
       for ist=1:8;

          StValueCount = StepDistributions{ist,2};
          StValueDist = L2ConversionFactor * StepDistributions{ist,3}(:,1); % step
values

          %fprintf('%5d ',StepDistributions{is,1});
          fprintf('%6d ',StValueCount);
          fprintf('%7.3f ',max(StValueDist));

          MinInd=StepCounts(ich,ist)+1;
          MaxInd=StepCounts(ich,ist)+StValueCount;
          StepStats{ich,ist}(MinInd:MaxInd) = StValueDist;

          StepCounts(ich,ist)=StepCounts(ich,ist)+StValueCount;

          %MaxStepValues(ich,is)=max(MaxStepValues(ich,is),max(StValueDist));


       end; % loop over select step counts

       fprintf('\n');
    end; % if file was found
```

```
   end % loop over channels
end % loop over file [names]


% Build cumulative step distributions and histograms

StepRange = cell(2,8); % bin definitions
StepHist = cell(2,8);  % histograms

if MakePlot1001, figure(1001); clf; end;% optional global step distribution plot

for ich=1:2 % double loop for step statistics
   for ist=1:8

      % clip off the unused pieces of StepStats
      StepStats{ich,ist}=StepStats{ich,ist}(1:StepCounts(ich,ist));

      % define the value range
      MaxV = 20*mean(StepStats{ich,ist}); % upper end
      DV = MaxV/1000;                  % bin size
      StepRange{ich,ist} = 0:DV:MaxV;     % range

      Missed=length(find(StepStats{ich,ist}>MaxV)); %  values left out

      % do the histogram
      W = histc(StepStats{ich,ist},StepRange{ich,ist}); % histogram
      StepHist{ich,ist} = cumsum(W)/StepCounts(ich,ist);% cumulative fractions

      if MakePlot1001, % optional cumulative step statistics plot
         figure(1001)
         subplot(2,4,ist)
         if ich==2, hold on; end;
         %stairs(StepRange{is,ich},log(W/DV)/log(10),'Color',[ich-1,0,2-ich]);
         %semilogy(StepRange{is,ich},W/DV,'Color',[ich-1,0,2-ich]);
         semilogx(StepRange{ich,ist},StepHist{ich,ist},'Color',[ich-1,0,2-ich]);
         if ich==2, % format and embellish
            xstring='Sq.dis (\mum)';
            tstring='CDF';
            nstring=sprintf('%d steps',SelectStepVals(ist));
            if SelectStepVals(ist)==1, nstring='1 step'; end;
            xf=0.0001; yf=0.925;
            ylim([0 1.2]);
            nmin=floor(log(DV)/log(10));
            nmax=ceil(log(MaxV)/log(10));
            xlim([10^nmin 10^nmax]);
            set(gca,'XTick',[10.^(nmin:nmax)]);
```

```
legend(sprintf('Ch1:%d',StepCounts(1,ist)),sprintf('Ch2:%d',StepCounts(2,ist)));
            legend('Location','SouthEast');
            FormatThisFigure;

         end
      end

   end
end

if MakePlot1001, FigureName=[OutputFolder 'CumulativeStepStats_' SpecialString];
SaveThisFigure; end;

%return % optional stop before building the individual scores
% done collecting step statistiscs


%%  Analysis of trajectory files

% Variables for point scoring
OverallPointCount = sum(FilePointCount,1);
OverallShortPoints                                                                    =
{zeros(OverallPointCount(1),1),zeros(OverallPointCount(2),1)};%cell(2,1);
OverallBiRanks    =    {zeros(OverallPointCount(1),3),zeros(OverallPointCount(2),3)};
%cell(2,1);
BiRankCount = zeros(2,1);

for ifile=1:FileCount % main analysis loop over trajectory files..

   SPC = cell(2,2);

   for ich=1:2      % ..and over channels

      File= [FolderPath ShortFileNames{ifile} Ext{ich}];

      % Length study is done for a subset of the files
      LENGTHSTUDY=false;
      if ~isempty(strfind(File,LS_SpecialString)), LENGTHSTUDY=true; end;

      fprintf('File:%3d ch.%1d : ',ifile,ich);

      load(File);
      fprintf('.');
      NTr = size(Trajectories,1);
      TrajectoryRanks = cell(NTr,1);
```

```
if ~exist('PixelSize','var'), % the pixel size should come with the data
   fprintf('    Domains **error**    No PixelSize found!\n   File: %s',File);
   return;
end;

L2ConversionFactor=PixelSize^2; % convert into physical units

% this will hold the output set of points
AllShortPoints = zeros(FilePointCount(ifile,ich),3);
ShortPointCount = 0;

AllPoints = zeros(FilePointCount(ifile,ich),3);


% collect the ranks for all points in this file/channel combination
CumulativeBiRanks = zeros(FilePointCount(ifile,ich),1);
PointCount = 0;

for TrNo = 1:NTr; % loop over trajectories

   ThisTrace = Trajectories{TrNo,3};

   ThisTrace(:,1:2) = PixelSize * ThisTrace(:,1:2);%convert the xy coordinates into
\mum

   TraceLength =   size(ThisTrace,1);% length of this trace


   %% find the forward and reverse step sizes for the trajectory %

   % these will hold the ranks for each number of steps
   FwRanks = zeros(length(SelectStepVals),TraceLength);
   BkRanks = zeros(length(SelectStepVals),TraceLength);

   for ist=1:length(SelectStepVals) % loop over the 8 step numbers

      FBSqDist = zeros(TraceLength,2);% forward and backward square
displacements

      StCount=SelectStepVals(ist); % step count for this step number

      for ip=1:TraceLength % loop over all entries in the trajectory

         % point StCount after the current frame
         IFw = find(ThisTrace(:,3)==ThisTrace(ip,3) + StCount,1);
```

```
            if ~isempty(IFw), FBSqDist(ip,1) = sum( (ThisTrace(ip,1:2) -
ThisTrace(IFw,1:2)).^2); end;

            % point StCount before the current frame
            IBk = find(ThisTrace(:,3)==ThisTrace(ip,3) - StCount,1);
            if ~isempty(IBk), FBSqDist(ip,2) = sum( (ThisTrace(ip,1:2) -
ThisTrace(IBk,1:2)).^2); end;

        end

        % convert the recorded distances into bins in Range
        FBRanks = min(ceil(FBSqDist/StepRange{ich,ist}(2)),...
          length(StepRange{ich,ist}));

        % convert the (nonzero) bin indices into percentages using AllRanks
        FBRanks(FBRanks(:,1)~=0,1)                                    =
StepHist{ich,ist}(FBRanks(FBRanks(:,1)~=0,1));
        FBRanks(FBRanks(:,2)~=0,2)                                    =
StepHist{ich,ist}(FBRanks(FBRanks(:,2)~=0,2));

        % put the info into the multi-step size rank structures
        FwRanks(ist,:) = FBRanks(:,1);
        BkRanks(ist,:) = FBRanks(:,2);


    end

    %% build a unique score for each point

    BothPresent=FwRanks>0 & BkRanks>0;

    BiRank=zeros(size(FwRanks));
    BiRank(BothPresent) = (FwRanks(BothPresent)+BkRanks(BothPresent))/2.0;


    WeightedRank = sum(BiRank(ScorePattern,:),1) ./
sum(BiRank(ScorePattern,:)~=0,1);
    %StDBiRank = sqrt( sum(BiRank.*BiRank,1) ./ sum(BiRank~=0,1) -
MeanBiRank .* MeanBiRank );


    TrajectoryRanks{TrNo} = WeightedRank';
    %TrajectoryRanks{TrNo,2} = StDBiRank;

    %% identify the points associated with shorter step sizes
```

```
      ShortPoints = find(WeightedRank>0 & WeightedRank < CutFactor)';

      AllShortPoints(ShortPointCount+(1:size(ShortPoints,1)),:) =...
         [ThisTrace(ShortPoints,1:2) TrNo*ones(size(ShortPoints,1),1)];

      ShortPointCount = ShortPointCount + size(ShortPoints,1);

      AllPoints(PointCount+(1:TraceLength),:) =...
         [ThisTrace(:,1:2) TrNo*ones(size(ThisTrace,1),1)];



      CumulativeBiRanks(PointCount+(1:TraceLength)) = TrajectoryRanks{TrNo};
      PointCount = PointCount + TraceLength;

   end

   fprintf('.');

   AllShortPoints=AllShortPoints(1:ShortPointCount,:);
   AllPoints=AllPoints(1:PointCount,:);

   if CutFactor>1, AllShortPoints=AllPoints; end;

   % keep this for channel to channel comparisons
   SPC{ich,1}=AllShortPoints;

   if MakePlot404, % individual histogram of scores
      figure(404)
      if ich==1, clf; else hold on; end;
      hr=0:0.01:1;
      w=hist(CumulativeBiRanks,hr);
      if ich==1, stairs(hr,w,'b'); else stairs(hr,w,'r'); end;

      if ich==1,
         ystring='Frequency';
         xstring='Score';
         title(sprintf('Scores %s',ShortFileNames{ifile}),'interpreter','none');
         FormatThisFigure;
      end

      if ich==2,
         legend('ch 1','ch 2');
         FormatThisFigure;
         FigureName=[OutputFolder ShortFileNames{ifile} '_Scores_both_Pt'
sprintf('%d',ScorePattern)];
```

```
        SaveThisFigure;
      end
    end

    CleanBiRanks=CumulativeBiRanks(~isnan(CumulativeBiRanks));

    MinInd=BiRankCount(ich)+1;
    MaxInd=BiRankCount(ich)+length(CleanBiRanks);

    OverallBiRanks{ich}(MinInd:MaxInd)=CleanBiRanks;
    BiRankCount(ich)=MaxInd;

    OutFileName=[OutputFolder ShortFileNames{ifile} sprintf('.ch%d',ich) OutputExt{ich}];
    save(OutFileName, 'ShortFileNames', 'Trajectories', 'PixelSize', 'StepRange', 'StepHist', 'WeightedRank', 'AllShortPoints');
    %save(OutFileName);

    %% Clustering (still inside the loop over files + channels)

    fprintf('.');
    %
    RootFileName = [OutputFolder ShortFileNames{ifile} sprintf('.ch%d',ich)];

    % list of distances to look at for clusters
    DVecLong = 0.001:0.001:1;

    % Clustering
    MyZ = linkage(AllShortPoints(:,1:2));
    T = cluster(MyZ,'cutoff',DVecLong,'criterion','distance');
    fprintf('.');


    % pull out the cluster arrangement with the chosen length scale
    BestInd=find(DVecLong>=MyLength{ich},1);
    BestLength=DVecLong(BestInd);
    Clusters=T(:,BestInd);

    % keep this for channel to channel comparisons
    SPC{ich,2}=Clusters;

    fprintf('.');

    % cell array to collect the cluster info
    % {<number of points>, <Nx2 array of points>}
    % for: (1) actual points, (2) tight contour, (3) fat contour
```

```
ClustersAndContours=cell(max(Clusters), 6);

% count the points in each cluster
CC=zeros(max(Clusters),1); % holds cluster sizes, should be replaced eventually
for IC=1:max(Clusters)
   CC(IC)=sum(Clusters==IC);
end

% This will collect area and perimeter info
% inside A / P, est. padding A /P , padded polyline A / P
ClusterAP = zeros(max(Clusters), 8);

%% Contour building loop over clusters %%
for ISCC=1:max(Clusters);
   %% Tight contour
   % Pick out a cluster
   Chozen = find(Clusters==ISCC);

   MyPoints = AllShortPoints(Chozen,1:2);

   % record the points of this cluster
   ClustersAndContours{ISCC,1}=CC(ISCC);
   ClustersAndContours{ISCC,2}=MyPoints(:,1:2);

   % Build the tight contour
   BestLength2=BestLength*1.01;

   [Contour, Diam] = TightContour(MyPoints, BestLength2);

   % record the tight contour - no repeat of the first point
   ClustersAndContours{ISCC,3}= size(Contour,1);
   ClustersAndContours{ISCC,4}= MyPoints(Contour,1:2);
   %% Build the fat contour

   MinAngle = pi / 6; % for spokes in circles added around corners

   FatContour = FatContourBuild(MyPoints(Contour,1:2), BestLength2, MinAngle);

   % record the fat contour - skip the repeat of the first point
   ClustersAndContours{ISCC,5}=size(FatContour,1)-1;
   ClustersAndContours{ISCC,6}=FatContour(1:end-1,:);
   %% Geometry - areas and perimeters

   % area inside the tight contour - assumed closed
   ClusterAP(ISCC,1) = polyarea(MyPoints(Contour,1), MyPoints(Contour,2));
```

```
        % length of inner perimeter
        ClusterAP(ISCC,2) = sum(sqrt(sum((MyPoints(Contour([2:end 1]),:) -
MyPoints(Contour,:)).^2,2)));

        % estimation of padding area (add rectangles + a circle)
        ClusterAP(ISCC,3) = ClusterAP(ISCC,1) + ClusterAP(ISCC,2) * BestLength2/2
+ pi*(BestLength2/2)^2;

        % estimation of padded contour length (innner perimeter + circle
        ClusterAP(ISCC,4) = ClusterAP(ISCC,2) + pi * BestLength2;

        % area of fat contour
        ClusterAP(ISCC,5) = polyarea(FatContour(:,1),FatContour(:,2));

        % perimeter of fat contour
        ClusterAP(ISCC,6) = sum(sqrt(sum((FatContour(2:end,:)-FatContour(1:end-
1,:)).^2,2)));

        % cluster diameter (no padding)
        ClusterAP(ISCC,7) = Diam;

        % cluster diameter (with padding)
        ClusterAP(ISCC,8) = Diam + BestLength;
    end
    fprintf('.');
    OutFileName=[OutputFolder ShortFileNames{ifile} sprintf('.ch%d',ich)
OutputExt{ich}];
    save(OutFileName, 'T', 'MyLength', 'BestLength', 'Clusters', 'ClustersAndContours',
'ClusterAP', '-append');
    fprintf('.');

    if FILEOUTPUT ,

        % metadata output
        % point count, perimeter, area, etc.
        fout = fopen([RootFileName
sprintf('_%dnm_meta.txt',floor(1000*BestLength))],'w');

        for ISC=1:max(Clusters)

            %if DEMO && ISC>12, break; end; % loop limited to the 12

            fprintf(fout,'%3d ', ISC); % cluster index

            fprintf(fout,'%4d ',ClustersAndContours{ISC,1});% original (slow) point count
```

```
            fprintf(fout,'%4d  ',ClustersAndContours{ISC,3});% point count on tight
contour
            fprintf(fout,'%4d  ',ClustersAndContours{ISC,5});% point count on padded
contour

            fprintf(fout,'%6.4f %6.4f %10.8f %6.4f %6.4f %6.4f %6.4f %6.4f\n',
ClusterAP(ISC,[1 3 5 2 4 6 7 8]));

        end

        fclose(fout);

        % contour output
        % polylines for each contour
        fout=fopen([RootFileName
sprintf('_%dnm_contours.txt',floor(1000*BestLength))],'w');

        % number of clusters on a separate line
        fprintf(fout,'%4d\n',max(Clusters));

        % write out the fat contours only

        for ISC=1:max(Clusters)
           % if DEMO && ISC>12, break; end; % loop limited to the 12
           % point counts from all contours on a single line
           fprintf(fout,'%4d', ClustersAndContours{ISC,5});
        end

        fprintf(fout,'\n');

        for ISC=1:max(Clusters)
           % if DEMO && ISC>12, break; end; % loop limited to the 12
           % pull out the contour data
           FatContourOut = ClustersAndContours{ISC,6};
           % x coordinates on a single line
           fprintf(fout,'%8.5f ',FatContourOut(:,1));
           fprintf(fout,'\n');
           % y cordinates on a single line
           fprintf(fout,'%8.5f ',FatContourOut(:,2));
           fprintf(fout,'\n');
        end

        fclose(fout);

    end;
    fprintf('.\n');
```

```
if LENGTHSTUDY,
   tic
   % loop over characteristic lengths
   APtot=zeros(length(DVecLong),8);
   for Ind=1:length(DVecLong)
      Clusters=T(:,Ind);
      APtot(Ind,5)=max(Clusters);
      DiamVals = zeros(max(Clusters),1);
      for ICl=1:max(Clusters)
         BestLength=DVecLong(Ind);
         MyPoints=AllShortPoints(Clusters==ICl,1:2);
         [Contour, DiamVals(ICl)] = TightContour(MyPoints, 1.01*BestLength);

         % area inside the tight contour - assumed closed
         AreaIn = polyarea(MyPoints(Contour,1), MyPoints(Contour,2));% area
inside the tight contour - assumed closed
         % length of inner perimeter
         PerimIn = sum(sqrt(sum((MyPoints(Contour([2:end 1]),:) -
MyPoints(Contour,:)).^2,2)));
         % estimation of padding area (add rectangles + a circle)
         AreaTot = AreaIn + PerimIn * BestLength/2 + pi*(BestLength/2)^2;
         % estimation of padded contour length (innner perimeter + circle
         PerimTot = PerimIn + pi * BestLength;

         APtot(Ind,1:4)=APtot(Ind,1:4)+[AreaIn, AreaTot, PerimIn, PerimTot];


      end

      APtot(Ind,6:8)=[mean(DiamVals),median(DiamVals),max(DiamVals)];

      fprintf('L=%3dnm NCl=%3d Ain=%6.3f Atot=%6.3f Pin=%5.3f Ptot=%5.3f
AvDiam=%5.3f\n',...
         floor(DVecLong(Ind)*1000), max(Clusters), APtot(Ind,1:4), APtot(Ind,6));

   end
   toc

   save(OutFileName, 'APtot','-append');

   if MakePlot2001, % length study plot

      figure(2001); clf;
      subplot(2,2,1) %
      plot(DVecLong,APtot(:,2));
```

```
            tstring='Aggregate Domain Area';
            xstring='L (\mum)'; ystring='\mum^2';
            FormatThisFigure;
            subplot(2,2,2) %
            plot(DVecLong,APtot(:,4));
            tstring='Aggregate Perimeter';
            xstring='L (\mum)'; ystring='\mum';
            FormatThisFigure;
            subplot(2,2,3) %
            plot(DVecLong,APtot(:,5));
            xlim([0 0.2]);
            tstring='Number of clusters';
            xstring='L (\mum)'; ystring='Count';
            FormatThisFigure;
            subplot(2,2,4) %
            plot(DVecLong,APtot(:,6),'b-');hold on;
            plot(DVecLong,APtot(:,7),'r-');
            legend('mean','median');
            tstring='Diameter';
            xstring='L (\mum)'; ystring='\mum';
            FormatThisFigure;

            FigureName=[OutputFolder                                    ShortFileNames{ifile}
sprintf('_ch%d_LengthStudy',ich)];
            SaveThisFigure;
          end

          %return
        end % LENGTHSTUDY

    end % end loop over channels

    % channel comparisons here
    Overlap_V001;

%     fout = fopen([OutputFolder ShortFileName...
%
sprintf('_Overlaps_%dnm_%dnm.txt',floor(1000*MyLength{1}),floor(1000*MyLength{
2}))]...
%         ,'w');
%

% save overlap info in both channel files
for ich=1:2
```

```
    OutFileName=[OutputFolder ShortFileNames{ifile} sprintf('.ch%d',ich) OutputExt{ich}];
      save(OutFileName,'TotalOverlap', 'OvlAreas', '-append');

end

end % end loop over files

if MakePlot405, % two channel summary histogram of global scores

   for ich=1:2

      OverallBiRanks{ich}=OverallBiRanks{ich}(1:BiRankCount(ich));
      figure(405)
      if ich==1, clf; else hold on; end;
      hr = 0:0.01:1;
      w=hist(OverallBiRanks{ich},hr);
      if ich==1, stairs(hr,w,'b'); else stairs(hr,w,'r'); end;
      xstring='Score';
      ystring='Counts';
      if ich==2,
         tstring='HA HRG Pattern: ';
         tstring = [tstring sprintf('%d',ScorePattern)];
         legend('ch1','ch2');
         FormatThisFigure;
         FigureName=sprintf('HA_HRG_%s_both_Pt',SpecialString);
         FigureName=[OutputFolder FigureName sprintf('%d',ScorePattern)];
         SaveThisFigure;
      end

   end

end

toc

return
```

ContourV004.m

```
%% Cluster calls and contour finding for 'short' points from an SPTData file %%

%% Development history / features
% ** Version 1 **
% * loads the result of the first two sets of processing steps performed
%   on a single movie file
% * requires a set of points, organized in a distance based cluster
```

```
%   structure, and a length parameter value
% * makes a plot of the points, colored by the cluster assignment
% * performs the contour finding for a set of selected clusters
% * plots the clusters andthe contours un a multi-plot
% * Contour finding algorithm:
%   - starts from the rightmost point
%     (could be any point on the convex hull)
%   - uses a reference direction
%     + initially the RD is the positive x
%   - Update steps:
%     + subsequently, RD is 60 degrees to the left of the direction
%       pointing from the current to the previous point
%     + identify all points within L of the current point
%     + calculates their angle w.r.to RD
%     + next point is the one with the smallest such angle
%   - Stopping criteria
%     + The initial point is hit
%       TODO: change this to crossing the initial segment in the same direction
%     + Number of steps exceeds nuber of points
%
% ** Version 2 **
%    Contour algorithm
%    * make sure it works for small clusters
%    * change stopping criterion to segment
%    * add expanded footprint (rectangles and circles)
%    * build polyline contour of expanded footprint
%
% ** Version 3 **
%
%    Contour algorithms
%    * more options for initial point, direction? -- later
%    * connect to convex hull somehow? - not necessary
%    * add universal self intersection check (uses polyxpoly() ) -- done
%    * convert to function -- done
%    * calculate area, perimeter, density, etc. -- done
%
%
%% Switches for behavior and output control %%

if ~exist('TOP12PLOTS','var'), TOP12PLOTS=true; end;% plots the top 12 clusters Fig.5002
if ~exist('CLUSTERCONTOURPLOTS','var'), CLUSTERCONTOURPLOTS=true; end;% Fig 5003
if ~exist('OVERLAYPLOTS','var'), OVERLAYPLOTS=false; end;% if true, used hold on; if false uses clf
if ~exist('FILEOUTPUT','var'), FILEOUTPUT=true; end; % outputs cluster info
```

```
if ~exist('LENGTHSTUDY','var'), LENGTHSTUDY=false; end; % loops over length values and stops

%RootFileName='ClusterDemoFile_Ch2';
%RootFileName=['Scratch/' ShortFileName];
RootFileName = 'Scratch/TestFile';

RootFileName = ShortFileName;

MyLength=0.035;%
ContourColor=[0 0 0];

% list of distances to look at for clusters
DVecLong = 0.001:0.001:1;

tic

% Clustering
MyZ = linkage(AllShortPoints(:,1:2));
MyT = cluster(MyZ,'cutoff',DVecLong,'criterion','distance');
T=MyT;

toc

if LENGTHSTUDY,
   tic
   % loop over characteristic lengths
   APtot=zeros(length(DVecLong),5);
   for Ind=1:length(DVecLong)
      Clusters=T(:,Ind);
      APtot(Ind,5)=max(Clusters);
      for ICl=1:max(Clusters)
         BestLength=DVecLong(Ind);
         MyPoints=AllShortPoints(Clusters==ICl,1:2);
         Contour = TightContour(MyPoints, 1.01*BestLength);

         % area inside the tight contour - assumed closed
         AreaIn = polyarea(MyPoints(Contour,1), MyPoints(Contour,2));% area inside the tight contour - assumed closed
         % length of inner perimeter
         PerimIn = sum(sqrt(sum((MyPoints(Contour([2:end 1]),:) - MyPoints(Contour,:)).^2,2)));
         % estimation of padding area (add rectangles + a circle)
         AreaTot = AreaIn + PerimIn * BestLength/2 + pi*(BestLength/2)^2;
         % estimation of padded contour length (innner perimeter + circle
         PerimTot = PerimIn + pi * BestLength;
```

```
        APtot(Ind,1:4)=APtot(Ind,1:4)+[AreaIn, AreaTot, PerimIn, PerimTot];

      end

      fprintf('L=%3dnm NCl=%3d Ain=%6.3f Atot=%6.3f Pin=%5.3f Ptot=%5.3f\n',...
        floor(DVecLong(Ind)*1000), max(Clusters), APtot(Ind,1:4));



  end
  toc

  figure(2001)

  FileName=[ShortFileName sprintf('_ch%d_LengthStudy',Channel)];

  save(FileName);

  return
end


% pull out the cluster arrangement with the chosen length scale
if exist('MyLength','var'),
  BestInd=find(DVecLong>=MyLength,1);
  BestLength=DVecLong(BestInd);
  Clusters=T(:,BestInd);
end




%% Preliminaries

% cell array to collect the cluster info
% {<number of points>, <Nx2 array of points>}
% for: (1) actual points, (2) tight contour, (3) fat contour
ClustersAndContours=cell(max(Clusters), 6);

% count the points in each cluster
CC=zeros(max(Clusters),1); % holds cluster sizes, should be replaced eventually
for IC=1:max(Clusters)
  CC(IC)=sum(Clusters==IC);
end
```

```
% find the top 12 clusters by size
Q = [[1:max(Clusters)]',CC];
Q = sortrows(Q,-2);
SpecialClusters = Q(1:12,1);

% This will collect area and perimeter info
% inside A / P, est. padding A /P , padded polyline A / P
ClusterAP = zeros(max(Clusters), 6);

if CLUSTERCONTOURPLOTS,
   figure(5003);
   if OVERLAYPLOTS, hold on;  else clf;  end;
   plot(AllShortPoints(:,1),AllShortPoints(:,2),'ro','MarkerSize',1);
end

%% Contour building loop over clusters %%
for ISCC=1:max(Clusters);

   %% Pick out a cluster %%
   Chozen = find(Clusters==ISCC);

   % this is nonempty if this cluster is on the list
   ISC=find(SpecialClusters==ISCC,1);


   MyPoints = AllShortPoints(Chozen,1:2);

   % record the points of this cluster
   ClustersAndContours{ISCC,1}=CC(ISCC);
   ClustersAndContours{ISCC,2}=MyPoints(:,1:2);

   % Build the tight contour
   BestLength2=BestLength*1.01;

   Contour = TightContour(MyPoints, BestLength2);


   % record the tight contour - no repeat of the first point
   ClustersAndContours{ISCC,3}= size(Contour,1);
   ClustersAndContours{ISCC,4}=MyPoints(Contour,1:2);

   % Build the fat contour

   MinAngle = pi / 6; % for spokes in circles added around corners
```

```
   FatContour = FatContourBuild(MyPoints(Contour,1:2), BestLength2, MinAngle);

   if TOP12PLOTS && ~isempty(ISC), % make a plot for this cluster
      figure(5002);
      subplot(4,3,ISC,'replace')

      % plot the points in the cluster
      plot(MyPoints(:,1),MyPoints(:,2),...
         'ro','MarkerSize',2,'MarkerFaceColor',[0.9 0.9 0],'LineWidth',0.5);

      hold on

      % plot the tight contour (close it)
      Inds=Contour([1:end,1]);% normal
      %StopInd=min(size(Contour,1),30);Inds=Contour(1:StopInd);%debug
      plot(MyPoints(Inds,1),MyPoints(Inds,2),...
         'bo-','MarkerSize',2,'LineWidth',0.5);

      % plot the fat contour (it is closed already)
      plot(FatContour(:,1),FatContour(:,2),'k-');

      % format and embellish
      axis equal
      tstring=sprintf('#%d (%d pts.)', ISCC, CC(ISCC));
      title(tstring);
      FormatThisFigure;% makes the output nicer
   end;


   % record the fat contour - skip the repeat of the first point
   ClustersAndContours{ISCC,5}=size(FatContour,1)-1;
   ClustersAndContours{ISCC,6}=FatContour(1:end-1,:);

   %% geometry - areas and perimeters

   % area inside the tight contour - assumed closed
   ClusterAP(ISCC,1) = polyarea(MyPoints(Contour,1), MyPoints(Contour,2));

   % length of inner perimeter
   ClusterAP(ISCC,2) = sum(sqrt(sum((MyPoints(Contour([2:end 1]),:) -
MyPoints(Contour,:)).^2,2)));

   % estimation of padding area (add rectangles + a circle)
   ClusterAP(ISCC,3) = ClusterAP(ISCC,1) + ClusterAP(ISCC,2) * BestLength2/2 +
pi*(BestLength2/2)^2;
```

```
   % estimation of padded contour length (innner perimeter + circle
   ClusterAP(ISCC,4) = ClusterAP(ISCC,2) + pi * BestLength2;

   % area of fat contour
   ClusterAP(ISCC,5) = polyarea(FatContour(:,1),FatContour(:,2));

   % perimeter of fat contour
   ClusterAP(ISCC,6)        =        sum(sqrt(sum((FatContour(2:end,:)-FatContour(1:end-
1,:)).^2,2)));



   if CLUSTERCONTOURPLOTS,

      figure(5003)
      hold on
      if  ClustersAndContours{ISCC,1}  >  0,  plot(FatContour(:,1),FatContour(:,2),'-
','Color',ContourColor); end;

      figure(1003)
      hold on
      if  ClustersAndContours{ISCC,1}  >  0,  plot(FatContour(:,1),FatContour(:,2),'-
','Color',ContourColor); end;
   end

end



if TOP12PLOTS,
   figure(5002);
   FigureName=[RootFileName sprintf('_%dnm_TopClusters',floor(1000*BestLength))];
   SaveThisFigure;
end;


if CLUSTERCONTOURPLOTS,
   figure(5003)
   %axis([22 23 15.5 16.5])
   FormatThisFigure;
   FigureName=[RootFileName '_AllContours'];
   %savefig(FigureName);
   SaveThisFigure;

end
```

```
if FILEOUTPUT ,

   % metadata output
   % point count, perimeter, area, etc.
   fout = fopen([RootFileName sprintf('_%dnm_meta.txt',floor(1000*BestLength))],'w');

   for ISC=1:max(Clusters)

      %if DEMO && ISC>12, break; end; % loop limited to the 12

      fprintf(fout,'%3d ', ISC); % cluster index

      fprintf(fout,'%4d ',ClustersAndContours{ISC,1});% original (slow) point count
      fprintf(fout,'%4d ',ClustersAndContours{ISC,3});% point count on tight contour
      fprintf(fout,'%4d ',ClustersAndContours{ISC,5});% point count on padded contour

      fprintf(fout,'%6.4f %6.4f %6.4f %6.4f %6.4f %6.4f\n', ClusterAP(ISC,[1 3 5 2 4
6]));

   end

   fclose(fout);

   % contour output
   % polylines for each contour
   fout=fopen([RootFileName
sprintf('_%dnm_contours.txt',floor(1000*BestLength))],'w');

   % number of clusters on a separate line
   fprintf(fout,'%4d\n',max(Clusters));

   % write out the fat contours only

   for ISC=1:max(Clusters)
      % if DEMO && ISC>12, break; end; % loop limited to the 12
      % point counts from all contours on a single line
      fprintf(fout,'%4d', ClustersAndContours{ISC,5});
   end

   fprintf(fout,'\n');

   for ISC=1:max(Clusters)
      % if DEMO && ISC>12, break; end; % loop limited to the 12
      % pull out the contour data
      FatContourOut=ClustersAndContours{ISC,6};
      % x coordinates on a single line
```

```
        fprintf(fout,'%8.5f ',FatContourOut(:,1));
        fprintf(fout,'\n');
        % y cordinates on a single line
        fprintf(fout,'%8.5f ',FatContourOut(:,2));
        fprintf(fout,'\n');
    end

    fclose(fout);

end;
```

APPENDIX B: Spatial Stochastic Model Program Script

ReceptorInfo Module

```
MODULE ReceptorInfo

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

    ! Species
    !
    !       (1) ErbB2
    !       (2) ErbB3
    !       (3) ErbB2.ErbB3
    !       (4) ErbB3.ErbB3
    !       (5) ErbB2.ErbB2

    ! Species Info
    INTEGER :: NumBaseSpecies
    INTEGER :: NumPhosStates
    INTEGER :: NumLigandStates
    INTEGER :: NumMonomerSpecies
    ! Species during simulation
    INTEGER, POINTER :: SpeciesCount(:)

    ! Species matrix f(monomer species, monomer species)
    INTEGER, POINTER :: SpeciesMatrix(:,:) ! 2,2
    ! Dimer Allowed to form f(species,ligandcount)
    LOGICAL, POINTER :: DimerForm(:,:) ! 5,3
    ! Diffusion Coefficients [=] um^2/s, f(species, phosphorylation)
    ! *** Parameters from Mara, lumped Monomer/Dimer together***
    !       Unphosphorylated Species
    DOUBLE PRECISION, POINTER :: DiffCoeff(:,:) ! 5,3
    ! Phosphorylation Rate [=] 1/s, f(species, phosphorylation)
    ! *** Parameters from Shankaran et al (2006) BiophysJ ***
    ! No Phosphorylation
    DOUBLE PRECISION, POINTER :: PhosRate(:,:) ! 5,3
    ! Dephosphorylation Rate [=] 1/s, f(species,phosphorylation)
    ! *** Parameters from Shankaran et al (2006) BiophysJ ***
    DOUBLE PRECISION, POINTER :: DePhosRate(:,:) ! 5,3
    ! Dimer Off Rate [=] 1/s, f(species,ligand)
    ! *** Parameters from Mara, unless noted ***
    ! No Ligand
    DOUBLE PRECISION, POINTER :: DimOffRate(:,:) ! 5,3
    ! Dimer Binding Radius [=] um, f(proposed species type, phosphorylation)
    ! Dimers forming with No Phos
```

```
DOUBLE PRECISION, POINTER :: BindRadius(:,:) ! 5,3
! Dimer UnBinding Radius [=] um, f(proposed species type, proposed ligand count)
! Dimers forming with No Ligand
DOUBLE PRECISION, POINTER :: UnBindRadius(:,:) ! 5,3
! Domain Escape Rate [=] 1/Frame, f(species)
! *** Shalini Data for now ***
DOUBLE PRECISION, POINTER :: EscapeRate(:) ! 5
! Receptor Flux Probability
DOUBLE PRECISION, POINTER :: RecFlip(:) ! 2
! Phosphorylation Multiplier
DOUBLE PRECISION, POINTER :: PhosMulti(:,:) ! 2,3
! Dephosphorylation Multiplier
DOUBLE PRECISION, POINTER :: DephosMulti(:,:) ! 2,3

Type Molecule

    ! Species [1-5]
    INTEGER :: OriginalSpecies
    INTEGER :: Species
    ! Ligand State [0 1]
    INTEGER :: Ligand
    ! Phos State [0 1]
    INTEGER :: Phosphate
    ! Position(2)
    DOUBLE PRECISION :: Position(2)
    ! Initial Position(2)
    DOUBLE PRECISION :: InitialPosition(2)
    ! Bound Partner
    INTEGER :: BoundBuddy
    ! Domain
    INTEGER :: Domain
    ! Active Receptor Receptor in dimer that is in active conformation
    INTEGER :: ActiveTail

END TYPE Molecule

TYPE MoleculeData

    ! Dimer Lifetime
    DOUBLE PRECISION :: DimerStart
    ! Phosphorylation Lifetime
    DOUBLE PRECISION :: PhosStart
    ! Last Jump Size [=] um
    DOUBLE PRECISION :: JumpSize(3)
    ! Tail has been phosphorylated in this dimer lifetime
```

```
        INTEGER :: PhosSuccess

    END TYPE MoleculeData

    !TYPE DomainLimits
!
!       DOUBLE PRECISION, POINTER :: xmin
!       DOUBLE PRECISION, POINTER :: xmax
!       DOUBLE PRECISION, POINTER :: ymin
!       DOUBLE PRECISION, POINTER :: ymax
!
!   END TYPE DomainLimits
!
!   TYPE DomainVertices
!
!       DOUBLE PRECISION, POINTER :: XCoord(:) => null()
!       DOUBLE PRECISION, POINTER :: YCoord(:) => null()
!
!   END TYPE DomainVertices

    TYPE DomainInfo

        DOUBLE PRECISION, POINTER :: xmin
        DOUBLE PRECISION, POINTER :: xmax
        DOUBLE PRECISION, POINTER :: ymin
        DOUBLE PRECISION, POINTER :: ymax
        DOUBLE PRECISION, POINTER :: XCoord(:) => null()
        DOUBLE PRECISION, POINTER :: YCoord(:) => null()

    END TYPE DomainInfo

    TYPE MembraneLimits

        DOUBLE PRECISION :: xlimmin
        DOUBLE PRECISION :: xlimmax
        DOUBLE PRECISION :: ylimmin
        DOUBLE PRECISION :: ylimmax

    END TYPE MembraneLimits

    TYPE SimulationData

        DOUBLE PRECISION :: Length ! Seconds
        DOUBLE PRECISION :: TimeStep ! Seconds
        DOUBLE PRECISION :: RTimeStep ! TimeStep per Receptor
        DOUBLE PRECISION :: CurrentTimeStep
```

```
        DOUBLE PRECISION :: FrameRate ! Frames per Second
        INTEGER :: NumberReceptors
        INTEGER, POINTER :: NumberDomains(:) ! => null()
        CHARACTER(80) :: OutputDirectory
        INTEGER, POINTER :: NumberVertices(:,:) ! => null()

    END TYPE SimulationData

    TYPE SpeciesData

        INTEGER :: Liganded(3)
        INTEGER :: Phosphorylated(3)

    END TYPE SpeciesData

    TYPE(Molecule), POINTER :: Receptor(:)
    TYPE(MoleculeData), POINTER :: ReceptorData(:)
    TYPE(DomainInfo), POINTER :: Domain(:,:)
    !TYPE(DomainLimits), POINTER :: Domain(:)
    !TYPE(DomainVertices), POINTER :: Vertex(:)
    TYPE(SpeciesData), POINTER :: Species(:)
    TYPE(MembraneLimits) :: Membrane
    TYPE(SimulationData) :: Simulation

END MODULE ReceptorInfo
```

BoundaryCondition Module

```
MODULE BCCheck

CONTAINS

    SUBROUTINE PeriodicBC(i,xjump,yjump)

        USE ReceptorInfo

        IMPLICIT NONE

        DOUBLE PRECISION, INTENT(IN) :: xjump, yjump! Proposed x jump,
Proposed y jump
        INTEGER, INTENT(IN) :: i ! current receptor
        DOUBLE PRECISION :: x, y ! proposed new position

        ! Calculate new receptor position
        x = xjump + Receptor(i)%Position(1)
        y = yjump + Receptor(i)%Position(2)
```

```
		!! Periodic Boundary Conditions
		! Check & Apply Periodic Boundary Condition
		! Check x move
		IF (x < Membrane%xlimmin) THEN ! Add width of box
			! Define X New Coordinate
			Receptor(i)%Position(1)=x+(Membrane%xlimmax-
Membrane%xlimmin)
		ELSE IF (x > Membrane%xlimmax) THEN ! Subtract width of box
			! Define X New Coordinate
			Receptor(i)%Position(1)=x-(Membrane%xlimmax-
Membrane%xlimmin)
		ELSE ! remains unchanged
			! Define X New Coordinate
			Receptor(i)%Position(1)=x
		END IF
		! Check y move
		IF (y < Membrane%ylimmin) THEN ! Add length of box
			! Define Y New Coordinate
			Receptor(i)%Position(2)=y+(Membrane%ylimmax-
Membrane%ylimmin)
		ELSE IF (y > Membrane%ylimmax) THEN ! Subtract length of box
			! Define Y New Coordinate
			Receptor(i)%Position(2)=y-(Membrane%ylimmax-
Membrane%ylimmin)
		ELSE ! remains unchanged
			! Define Y New Coordinate
			Receptor(i)%Position(2)=y
		END IF
			! No periodic condition applied to allow for free diffusion

	END SUBROUTINE PeriodicBC

	SUBROUTINE ReflectiveBC(i,xjump,yjump)

		! Condition: Only for receptors in domains, does not work for free
receptors

		USE ReceptorInfo

		IMPLICIT NONE

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

		IF (Receptor(i)%Domain == 0) THEN
```

```
                WRITE(*,*) 'ERROR : FREE RECEPTOR PASSED TO
ReflectiveBC Subroutine'
            END  IF

            ! Check & Apply Reflective Boundary Condition
            x=Receptor(i)%Position(1)+xjump
            y=Receptor(i)%Position(2)+yjump
            !WRITE(*,*) 'x = ', x, 'y = ', y
            ! Check x move
            IF                              (x                          <
Domain(Receptor(i)%Domain,Receptor(i)%OriginalSpecies)%xmin) THEN
                ! Define X New Coordinate

        Receptor(i)%Position(1)=2*Domain(Receptor(i)%Domain,Receptor(i)%Original
Species)%xmin&
                &-xjump-Receptor(i)%Position(1)
            ELSE                    IF                  (x                  >
Domain(Receptor(i)%Domain,Receptor(i)%OriginalSpecies)%xmax) THEN
                ! Define X New Coordinate

        Receptor(i)%Position(1)=2*Domain(Receptor(i)%Domain,Receptor(i)%Original
Species)%xmax&
                &-xjump-Receptor(i)%Position(1)
            END IF
            ! Check y move

            IF                              (y                          <
Domain(Receptor(i)%Domain,Receptor(i)%OriginalSpecies)%ymin) THEN
                ! Define Y New Coordinate

        Receptor(i)%Position(2)=2*Domain(Receptor(i)%Domain,Receptor(i)%Original
Species)%ymin&
                &-yjump-Receptor(i)%Position(2)
            ELSE                    IF                  (y                  >
Domain(Receptor(i)%Domain,Receptor(i)%OriginalSpecies)%ymax) THEN
                ! Define Y New Coordinate

        Receptor(i)%Position(2)=2*Domain(Receptor(i)%Domain,Receptor(i)%Original
Species)%ymax&
                &-yjump-Receptor(i)%Position(2)
            END IF

        END SUBROUTINE ReflectiveBC

        SUBROUTINE DomainCheck(i,xjump,yjump,domainnum,DomainChange)
```

```
        USE ReceptorInfo

        IMPLICIT NONE

        INTEGER, INTENT(IN) :: i ! Current receptor
        INTEGER, INTENT(OUT) :: domainnum ! domain where proposed coordinates
fall, if 0 not in a domain
        DOUBLE PRECISION, INTENT(IN) :: yjump, xjump
        INTEGER :: k, numdomains, j, kk, nverts, ReceptorDomain, c, jjj ! Counter
        DOUBLE PRECISION :: x, y ! proposed new location
        LOGICAL, INTENT(OUT) :: DomainChange
        LOGICAL                                                                 ::
xmincheck(Simulation%NumberDomains(Receptor(i)%OriginalSpecies))
        LOGICAL                                                                 ::
ymincheck(Simulation%NumberDomains(Receptor(i)%OriginalSpecies))
        LOGICAL                                                                 ::
xmaxcheck(Simulation%NumberDomains(Receptor(i)%OriginalSpecies))
        LOGICAL                                                                 ::
ymaxcheck(Simulation%NumberDomains(Receptor(i)%OriginalSpecies))

        x=Receptor(i)%Position(1) + xjump
        y=Receptor(i)%Position(2) + yjump
        DomainChange = .FALSE.
        ReceptorDomain = Receptor(i)%Domain
        numdomains = Simulation%NumberDomains(Receptor(i)%OriginalSpecies)
        !ALLOCATE(xmincheck(Simulation%NumberDomains),
xmaxcheck(Simulation%NumberDomains))
        !ALLOCATE(ymincheck(Simulation%NumberDomains),
ymaxcheck(Simulation%NumberDomains))
        ! Check where the new coordinate falls in the simulation space
        domainnum = 0 !Receptor(i)%Domain
        IF (numdomains /= 0) THEN
                !WRITE(*,*) 'Check if Receptor is in a Domain. Receptor Species =',
Receptor(i)%OriginalSpecies, 'numdomains =',  numdomains
                ! Create boolean vectors for each domain edge
                DO jjj = 1, numdomains
                        xmincheck               =               x               >=
Domain(jjj,Receptor(i)%OriginalSpecies)%xmin
                        xmaxcheck               =               x               <=
Domain(jjj,Receptor(i)%OriginalSpecies)%xmax
                        ymincheck               =               y               >=
Domain(jjj,Receptor(i)%OriginalSpecies)%ymin
                        ymaxcheck               =               y               <=
Domain(jjj,Receptor(i)%OriginalSpecies)%ymax
                END DO
                !WRITE(*,*) x, y
```

```
                !WRITE(*,*) xmincheck, xmaxcheck, ymincheck, ymaxcheck
                DO k = 1,numdomains
                        IF (xmincheck(k) .AND. xmaxcheck(k) .AND. ymincheck(k)
.AND. ymaxcheck(k)) THEN
                                !WRITE(*,*) 'Receptor MAY be in a domain'
                                domainnum = k
                                ! Check if receptor is actually in polygon
                                nverts                                          =
Simulation%NumberVertices(k,Receptor(i)%OriginalSpecies)
                                kk = nverts - 1
                                j = 1
                                c = 0
                                DO WHILE (j < nverts)

                                        ! ,Receptor(i)%OriginalSpecies

                                        !IF    (((Vertex(k)%YCoord(j)    >    y)    .NEQV.
(Vertex(k)%YCoord(kk)>y)) .AND. &
!                                       (x<(Vertex(k)%XCoord(kk)-
Vertex(k)%XCoord(j))*(y-Vertex(k)%YCoord(j))&
!                                       /(Vertex(k)%YCoord(kk)-
Vertex(k)%YCoord(j))+Vertex(k)%XCoord(j))) THEN
                                        IF
(((Domain(k,Receptor(i)%OriginalSpecies)%YCoord(j) > y) .NEQV. &

        &(Domain(k,Receptor(i)%OriginalSpecies)%YCoord(kk)>y)) .AND. &

        (x<(Domain(k,Receptor(i)%OriginalSpecies)%XCoord(kk)-&

        &Domain(k,Receptor(i)%OriginalSpecies)%XCoord(j))*&
                                        &(y-
Domain(k,Receptor(i)%OriginalSpecies)%YCoord(j))&

        /(Domain(k,Receptor(i)%OriginalSpecies)%YCoord(kk)-&

        &Domain(k,Receptor(i)%OriginalSpecies)%YCoord(j))+&

        &Domain(k,Receptor(i)%OriginalSpecies)%XCoord(j))) THEN
                                                c = c + 1
                                        END IF
                                        kk = j
                                        j = j + 1
                                END DO
                                IF (MOD(c,2) /= 0) THEN
                                        !WRITE(*,*) 'Receptor IS in a domain'
                                        IF (domainnum /= ReceptorDomain) THEN
```

```
                                        DomainChange = .True.
                                    END IF
                                ELSE
                                    !WRITE(*,*) 'Receptor IS NOT in a domain'
                                    domainnum = 0
                                END IF
                            END   IF
                        END DO
                END IF
                IF (domainnum == 0 .AND. domainnum /= ReceptorDomain) THEN
                        DomainChange = .True.
                END IF
                !WRITE(*,*) 'Domain Change?', DomainChange, 'New Domain', domainnum,
'Old Domain', Receptor(i)%Domain

                END SUBROUTINE DomainCheck

                SUBROUTINE DomainEscape(EscapeProb,i,xjump,yjump,domainnum)

                        ! Condition: Only for receptors in domains, does not work for free
receptors

                        USE ReceptorInfo
                        USE mtmod

                        IMPLICIT NONE

                        DOUBLE PRECISION, INTENT(IN) :: EscapeProb, xjump, yjump
                        INTEGER, INTENT(IN) :: i, domainnum ! Current receptor, anticipated
domain
                        INTEGER :: domainnum2
                        LOGICAL :: DomainChange2

                        DomainChange2 = .False.

                        IF (Receptor(i)%Domain == 0 .AND. EscapeProb < 1) THEN
                                WRITE(*,*) 'ERROR : FREE RECEPTOR PASSED TO
DomainEscape Subroutine'
                        END   IF

                        !IF (Receptor(i)%Species == 3) THEN
!                               IF      (Receptor(i)%Domain     >       0       .AND.
Receptor(Receptor(i)%BoundBuddy)%Domain > 0) THEN
!                                       WRITE(*,*) 'Dimer with Receptors in their own domains',
Receptor(i)%OriginalSpecies,&
!                                       & Receptor(Receptor(i)%BoundBuddy)%OriginalSpecies
```

```
!                       END IF
!               END IF

                IF (EscapeProb >= grnd()) THEN ! Escape!
                        IF (EscapeProb /= 1) THEN
                                !WRITE(*,*) 'ESCAPE! Successful Domain Change'
                        END IF
                        IF (domainnum == 0) THEN ! Moving to free membrane, check
periodic BC
                                !WRITE(*,*) 'Periodic Check'
                                CALL PeriodicBC(i,xjump,yjump)
                        ELSE ! Moving to another domain
                                !WRITE(*,*) 'Auto Assign'
                                Receptor(i)%Position(1)=Receptor(i)%Position(1)+xjump
                                Receptor(i)%Position(2)=Receptor(i)%Position(2)+yjump
                                Receptor(i)%Domain=domainnum
                        END IF

                ELSE ! Reflect back into domain
                        !WRITE(*,*) 'Reflective Check'
                        CALL ReflectiveBC(i,xjump,yjump)
                END IF

                IF (Receptor(i)%Species == 3) THEN ! Heterodimer

                                ! Check of opposite species leaves its own domain
                                CALL
DomainCheck(Receptor(i)%BoundBuddy,xjump,yjump,domainnum2,DomainChange2)

                                ! Update bound receptors position

        Receptor(Receptor(i)%BoundBuddy)%Position(1)=Receptor(i)%Position(1)+xju
mp

        Receptor(Receptor(i)%BoundBuddy)%Position(2)=Receptor(i)%Position(2)+yju
mp

                                ! Update bound buddy domain information
                                IF (DomainChange2) THEN
                                        !WRITE(*,*) 'Dimer with Receptors in their own
domains', Receptor(i)%OriginalSpecies,&
                                        !&
Receptor(Receptor(i)%BoundBuddy)%OriginalSpecies
                                        !WRITE(*,*)      'New      Domain      1',
Receptor(i)%Domain, 'Domain 2', Receptor(Receptor(i)%BoundBuddy)%Domain
```

```
        Receptor(Receptor(i)%BoundBuddy)%Domain=domainnum2

                                        !WRITE(*,*) '-----------------------------^^^^^^^------
------------'
                                        !WRITE(*,*) 'Domain Change for BoundBuddy?',
DomainChange2
                                        !WRITE(*,*) 'domainnum =', domainnum,
'domainnum2 =', domainnum2
                                        !WRITE(*,*) 'New Domain 1',
Receptor(i)%Domain, 'New Domain 2', Receptor(Receptor(i)%BoundBuddy)%Domain
                                        !WRITE(*,*) '-----------------------------------------
---------'
                                END IF

                        END IF

        END SUBROUTINE DomainEscape

END MODULE BCCheck
```

ReceptorReactions Module

```
MODULE ReceptorReactions

CONTAINS

        ! Diffusion (Dimerization Possible)
        SUBROUTINE ReceptorDiffuse(i,Reaction)

                USE ReceptorInfo
                USE mtmod
                USE BCCheck

                IMPLICIT NONE

                INTEGER, INTENT(IN) :: i
                DOUBLE PRECISION :: r1, r2, w1, w2, xjump, yjump, DiffSTD,
EscapeProb
                INTEGER :: LigandCount, PhosphateCount, domainnum, domainnum2
                LOGICAL :: DomainChange, DomainChange2, Dimer, Reaction

                ! Calculate Ligand count and phosphate count of current receptor and its
partner (if applicable)
                Reaction = .False.
                Dimer = .False.
                DomainChange2 = .False.
```

```
                IF (Receptor(i)%BoundBuddy > 0) THEN ! Dimer
                        Dimer = .True.
                        LigandCount = Receptor(i)%Ligand +
Receptor(Receptor(i)%BoundBuddy)%Ligand + 1
                        PhosphateCount = Receptor(i)%Phosphate +
Receptor(Receptor(i)%BoundBuddy)%Phosphate + 1
                ELSE ! Monomer
                        LigandCount = Receptor(i)%Ligand + 1
                        PhosphateCount = Receptor(i)%Phosphate + 1
                END IF

                ! Calculate Escape Probability
                EscapeProb =
EscapeRate(Receptor(i)%Species)*Simulation%FrameRate*Simulation%TimeStep

                ! Calculate Diffusion Deviation, DiffCoeff(Species,Phosphate)
                ! ***increase phosphate count by 1 due to array indexing starting at 1 not
0***

        DiffSTD=sqrt(2*DiffCoeff(Receptor(i)%Species,PhosphateCount)*Simulation%
TimeStep)

                !!! 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
                xjump=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)
                yjump=r2*w2

                ! Account for diffusion coefficient based on species type
                yjump=yjump*DiffSTD
                xjump=xjump*DiffSTD

                !! Add new normaly distributed distance to previous distance
                ! Define Move Distance
                ReceptorData(i)%JumpSize(1)=xjump ! dx(j,i)=xjump
                ReceptorData(i)%JumpSize(2)=yjump ! dy(j,i)=yjump
                ! Overall Distance Interval Calculation
                ReceptorData(i)%JumpSize(3)=sqrt(yjump**2+xjump**2)                !
d(j,i)=sqrt(yjump**2+xjump**2)
                ! Assign move distances to bound receptor if necessary
                IF (Dimer) THEN

        ReceptorData(Receptor(i)%BoundBuddy)%JumpSize(:)=ReceptorData(i)%Jump
Size(:)
                END IF

                ! Determine if the particle leaves the domain or stays in the domain
                CALL DomainCheck(i,xjump,yjump,domainnum,DomainChange)

                IF (DomainChange) THEN ! Changing domains
                        ! Check if leaving or entering domain
                        IF (Receptor(i)%Domain > 0) THEN ! Leaving domain
                                ! Check if receptor escapes, if yes move to new domain
                                CALL
DomainEscape(EscapeProb,i,xjump,yjump,domainnum)
                        ELSE ! Free receptor moving to domain, always allowed
                                ! execute with escape prob = 1
                                CALL DomainEscape(1.0d0,i,xjump,yjump,domainnum)
                        END IF
                ELSE ! Not Changing domains
                        ! execute with escape prob = 1 just updates position or calls
periodic if needed
                        CALL DomainEscape(1.0d0,i,xjump,yjump,domainnum)
```

```
                END IF

                IF (.NOT. Dimer) THEN ! Monomer, Check for dimerization
                        CALL BindReaction(i,Dimer)
                        Reaction = Dimer
                END IF

                IF (Dimer .AND. Receptor(i)%Species > 3) THEN ! Homodimers
                        IF (Receptor(i)%BoundBuddy == 0) THEN
                                WRITE(*,*) 'ERROR : Monomer being treated as Dimer in
ReceptorDiffuse Subroutine'
                        END IF
                        ! Update boundbuddy info

        Receptor(Receptor(i)%BoundBuddy)%Position(1)=Receptor(i)%Position(1)

        Receptor(Receptor(i)%BoundBuddy)%Position(2)=Receptor(i)%Position(2)

        Receptor(Receptor(i)%BoundBuddy)%Domain=Receptor(i)%Domain

                END IF

        END SUBROUTINE ReceptorDiffuse

        ! Dimerization Reaction
        SUBROUTINE BindReaction(i,Reaction)

                ! CONDITION: Can only be called for monomer species!!!

                USE ReceptorInfo
                USE mtmod

                IMPLICIT NONE

                INTEGER, INTENT(IN) :: i ! current particle
                LOGICAL, INTENT(OUT) :: Reaction
                LOGICAL :: monomercheck(Simulation%NumberReceptors), DimerPoss,
PhosCheck
                INTEGER :: PickedSpecies, Phosphate(Simulation%NumberReceptors),
LigandCount(Simulation%NumberReceptors)
                INTEGER :: NewSpecies, k
                DOUBLE PRECISION :: x_POI, y_POI,
distsq(Simulation%NumberReceptors), rannum
                Reaction = .FALSE.
                PhosCheck = .TRUE.
                ! Error check
```

```
            IF (Receptor(i)%BoundBuddy > 0) THEN
                  WRITE(*,*) 'ERROR : DIMER SPECIES PASSED TO
BindReaction Subroutine'
            END  IF
            x_POI=Receptor(i)%Position(1)
            y_POI=Receptor(i)%Position(2)
            PickedSpecies = Receptor(i)%Species ! Always a monomer

            ! Calculate distance between picked receptor and other receptors
            distsq=sqrt((Receptor%Position(1)-x_POI)**2+(Receptor%Position(2)-
y_POI)**2)
            ! Determine monomers
            monomercheck = Receptor%Species <= SIZE(SpeciesMatrix,1)
            ! Calculate proposed phosphate & species, Add 1 for index counting
starting at 1, 0 phosphates accessed by variable(1)
            Phosphate = Receptor(i)%Phosphate+Receptor%Phosphate + 1
            LigandCount = Receptor(i)%Ligand+Receptor%Ligand + 1
            DO k = 1,Simulation%NumberReceptors
                  IF (monomercheck(k)) THEN
                        NewSpecies                                    =
SpeciesMatrix(PickedSpecies,Receptor(k)%Species)
                        DimerPoss = DimerForm(NewSpecies,LigandCount(k))

                        IF      (DimerPoss      .AND.      distsq(k)      <=
BindRadius(NewSpecies,Phosphate(k)) .AND. k/=i) THEN

                              rannum = 0
                              IF (LigandCount(k) < 3) THEN
                                    rannum = grnd()
                              END IF

                              IF (rannum <= RecFlip(PickedSpecies)) THEN !
Check conformation of first monomer
                                    rannum = 0
                                    IF (LigandCount(k) == 1) THEN
                                          rannum = grnd()
                                    END IF

                                    IF                (rannum                <=
RecFlip(Receptor(k)%Species)) THEN ! Check conformation of second monomer

                                          Reaction = .True.

      SpeciesCount(PickedSpecies)=SpeciesCount(PickedSpecies)-1

      SpeciesCount(Receptor(k)%Species)=SpeciesCount(Receptor(k)%Species)-1
```

```
                                                    ! Record Dimer Start Time
                                                    Receptor(i)%Species = NewSpecies
                                                    Receptor(k)%Species = NewSpecies
                                                    ReceptorData(i)%DimerStart        =
Simulation%CurrentTimeStep
                                                    ReceptorData(k)%DimerStart        =
Simulation%CurrentTimeStep
                                                    Receptor(i)%BoundBuddy=k
                                                    Receptor(k)%BoundBuddy=i

        SpeciesCount(PickedSpecies)=SpeciesCount(PickedSpecies)+1
                                                    ! Pick active receptor of dimer
                                                    rannum = grnd()
                                                    IF (rannum <= .5) THEN
                                                            Receptor(i)%ActiveTail=i
                                                            Receptor(k)%ActiveTail=i
                                                    ELSE
                                                            Receptor(i)%ActiveTail=k
                                                            Receptor(k)%ActiveTail=k
                                                    END IF
                                                END IF
                                            END IF
                                            RETURN
                                        END IF
                                    END IF
                                END DO

        END SUBROUTINE BindReaction

! Dimer dissociation Reaction
        SUBROUTINE DissociationReaction(i)

                USE ReceptorInfo
                USE mtmod
                USE BCCheck

                IMPLICIT NONE

                INTEGER, INTENT(IN) :: i
                DOUBLE PRECISION :: placeangle, xjump, yjump, EscapeProb
                INTEGER :: LigandCount, PhosphateCount, domainnum, k
                LOGICAL :: DomainChange


                ! We know dissociation will occur, need to sort out data information
```

```
        SpeciesCount(Receptor(i)%Species)=SpeciesCount(Receptor(i)%Species)-1
                ! Calculate new postion for moving (unbinding) receptor
                ! Randomly pick position to place chosen part
                placeangle=2*Pi*grnd() ! Pick a number between 0 and 2*Pi
                ! Calculate LigandCount
                LigandCount = Receptor(i)%Ligand +
Receptor(Receptor(i)%BoundBuddy)%Ligand + 1
                PhosphateCount = Receptor(i)%Phosphate +
Receptor(Receptor(i)%BoundBuddy)%Phosphate + 1
                ! Separate Receptors
                xjump =
cos(placeangle)*UnBindRadius(Receptor(i)%Species,LigandCount)
                yjump =
sin(placeangle)*UnBindRadius(Receptor(i)%Species,LigandCount)
                IF (xjump == 0) THEN
                        WRITE(*,*) 'ERROR in DissociationReaction'
                END IF
                ! Check if moving receptor changes domains and update position
accordingly
                IF (Receptor(Receptor(i)%BoundBuddy)%Domain > 0) THEN ! Check if
moves from domain
                        CALL
DomainCheck(Receptor(i)%BoundBuddy,xjump,yjump,domainnum,DomainChange)
                        IF (DomainChange) THEN
                                ! Calculate Escape Probability
                                EscapeProb =
EscapeRate(Receptor(Receptor(i)%BoundBuddy)%OriginalSpecies)*Simulation%Frame
Rate*Simulation%TimeStep
                                CALL
DomainEscape(EscapeProb,Receptor(i)%BoundBuddy,xjump,yjump,domainnum)
                        ELSE
                                CALL
DomainEscape(1.0d0,Receptor(i)%BoundBuddy,xjump,yjump,domainnum)
                        END IF
                END IF

                ! Record Dimer lifetime
                ! Receptor 1 receptor 2 Dimer Start, Dimer End, Difference
                WRITE(4,*)
i,Receptor(i)%BoundBuddy,Receptor(i)%Species,LigandCount,PhosphateCount,Recepto
rData(i)%DimerStart, &

        &Simulation%CurrentTimeStep,Simulation%CurrentTimeStep-
ReceptorData(i)%DimerStart
```

```
                ! Update both receptors info back to monomer status

        SpeciesCount(Receptor(i)%OriginalSpecies)=SpeciesCount(Receptor(i)%Origina
lSpecies)+1

        SpeciesCount(Receptor(Receptor(i)%BoundBuddy)%OriginalSpecies)=SpeciesC
ount(Receptor(Receptor(i)%BoundBuddy)%OriginalSpecies)+1
                Receptor(Receptor(i)%BoundBuddy)%Species                =
Receptor(Receptor(i)%BoundBuddy)%OriginalSpecies
                Receptor(Receptor(i)%BoundBuddy)%BoundBuddy = 0
                ReceptorData(Receptor(i)%BoundBuddy)%DimerStart = 0
                ReceptorData(Receptor(i)%BoundBuddy)%PhosSuccess = 0
                ReceptorData(i)%PhosSuccess = 0
                Receptor(i)%ActiveTail = 0
                Receptor(Receptor(i)%BoundBuddy)%ActiveTail = 0
                Receptor(i)%Species = Receptor(i)%OriginalSpecies
                Receptor(i)%BoundBuddy = 0
                ReceptorData(i)%DimerStart = 0


        END SUBROUTINE DissociationReaction

        ! Phosphorylation
        SUBROUTINE Phosphorylate(i)

                USE ReceptorInfo
                USE mtmod

                IMPLICIT NONE

                INTEGER, INTENT(IN) :: i ! Particle number,

                IF (Receptor(i)%Species <= SIZE(SpeciesMatrix,1)) THEN
                        WRITE(*,*) 'ERROR : in Phosphorylate subroutine, monomer
species'
                END IF
                !! Symmetric Model !!
                ! update phosphate
                !Receptor(i)%Phosphate = 1
                !Receptor(Receptor(i)%BoundBuddy)%Phosphate = 1
                !IF (Receptor(i)%Species == 4) THEN
                        WRITE(*,*) 'In Function: PHOSPHORYLATE! Species: ',
Receptor(i)%Species!, 'Ligand Count: ', LigandCount, 'Phos Count: ', PhosphateCount
                        !WRITE(*,*) 'Species = ', Receptor(i)%Species,'UnbindProb =',
UnbindProb, 'PhosProb =', PhosProb, 'DephosProb =', DephosProb
```

```
                    WRITE(*,*) 'Ligand Count (+1 for indexing)=',
Receptor(i)%Ligand + Receptor(Receptor(i)%BoundBuddy)%Ligand + 1
                    WRITE(*,*) 'PhosState of ActiveTail =',
Receptor(Receptor(i)%ActiveTail)%Phosphate
                !END IF
                !! Asymmetric Model
                Receptor(Receptor(i)%ActiveTail)%Phosphate = 1
                ! update phos start
                ReceptorData(Receptor(i)%ActiveTail)%PhosStart                =
Simulation%CurrentTimeStep
                ReceptorData(Receptor(i)%ActiveTail)%PhosSuccess = 1
                !ReceptorData(Receptor(i)%BoundBuddy)%PhosStart                =
Simulation%CurrentTimeStep

                ! Record phosphorylation reaction information
                ! Active Receptor (one phosphorylated), Active Receptor Original
Species, Receiver Receptor, Receiver Phos state, timestamp
                WRITE(8,*)                                    Receptor(i)%ActiveTail,
Receptor(Receptor(i)%ActiveTail)%OriginalSpecies,&

        &Receptor(Receptor(i)%ActiveTail)%BoundBuddy,Receptor(Receptor(Receptor(
i)%ActiveTail)%BoundBuddy)%OriginalSpecies,&

        &Receptor(Receptor(Receptor(i)%ActiveTail)%BoundBuddy)%Phosphate,
Simulation%CurrentTimeStep

        END SUBROUTINE Phosphorylate

        ! Dephosphorylation
        SUBROUTINE DePhosphorylate(i)

        ! Dephosphorylation Subroutine. Only dephosphorylates the picked receptor. If
the
        ! receptor is bound to another receptor, the bound receptor is left alone.
                USE ReceptorInfo
                USE mtmod

                IMPLICIT NONE

                INTEGER, INTENT(IN) :: i

                Receptor(i)%Phosphate = 0

                ! Record Phos lifetime
                ! Receptor 1 receptor 2 Phos Start, Phos End, Difference
```

```
                WRITE(9,*)
i,Receptor(i)%BoundBuddy,Receptor(i)%Species,ReceptorData(i)%PhosStart, &

        &Simulation%CurrentTimeStep,Simulation%CurrentTimeStep-
ReceptorData(i)%PhosStart

                ReceptorData(i)%PhosStart = 0

        END SUBROUTINE DePhosphorylate

END MODULE ReceptorReactions
```

Main Spatial Stochastic Simulation Module

```
PROGRAM ErbB23_Sim

        ! Use mtmod to generate random number
        USE mtmod
        USE ReceptorInfo
        USE BCCheck
        USE ReceptorReactions

        IMPLICIT NONE

        INTEGER :: i, k, seed, LigandCount, PhosphateCount, datacutcount, datacut
        INTEGER :: framecount, frames, DomainLimitsFile(2), DomainVertFile(2)
        LOGICAL :: Reaction
        INTEGER*8 :: NMoves, RMoves, j
        CHARACTER*200 fnstring, fnstring2, fnstring3, fnstring4, fnstring5, fnstring6,
fnstring7 ! Filename string
        DOUBLE PRECISION :: UnbindProb, PhosProb, DephosProb, randnum

        WRITE(*,*) 'Importing Parameter Data'

        OPEN(4,file='ParameterInput')

        READ(4,*) NumBaseSpecies, NumPhosStates, NumLigandStates,
NumMonomerSpecies

        ALLOCATE(SpeciesMatrix(NumMonomerSpecies,NumMonomerSpecies),Dime
rForm(NumBaseSpecies,NumLigandStates))
        ALLOCATE(DiffCoeff(NumBaseSpecies,NumPhosStates),PhosRate(NumBaseS
pecies,NumPhosStates))
        ALLOCATE(DePhosRate(NumBaseSpecies,NumPhosStates),DimOffRate(Num
BaseSpecies,NumLigandStates))
```

```
        ALLOCATE(BindRadius(NumBaseSpecies,NumPhosStates),UnBindRadius(Nu
mBaseSpecies,NumLigandStates))
        ALLOCATE(EscapeRate(NumBaseSpecies),RecFlip(NumMonomerSpecies),Spe
ciesCount(NumBaseSpecies))
        ALLOCATE(PhosMulti(NumMonomerSpecies,NumPhosStates),
DephosMulti(NumMonomerSpecies,NumPhosStates))

        SpeciesCount(:) = 0
        ! Assign input values to parameters
        DO k = 1,NumPhosStates
                READ(4,*) DiffCoeff(:,k)
        END DO
        DO k = 1,NumPhosStates
                READ(4,*) PhosRate(:,k)
        END DO
        DO k = 1,NumPhosStates
                READ(4,*) DePhosRate(:,k)
        END DO
        DO k = 1,NumLigandStates
                READ(4,*) DimOffRate(:,k)
        END DO
        DO k = 1,NumPhosStates
                READ(4,*) BindRadius(:,k)
        END DO
        DO k = 1,NumLigandStates
                READ(4,*) UnBindRadius(:,k)
        END DO
        READ(4,*) EscapeRate(:)
        READ(4,*) RecFlip(:)
        DO k = 1,NumMonomerSpecies
                Read(4,*) SpeciesMatrix(:,k)
        END DO
        DO k = 1,NumLigandStates
                READ(4,*) DimerForm(:,k)
                !WRITE(*,*) DimerForm(:,k)
        END DO
!       DO k = 1,NumPhosStates
!               READ(4,*) PhosMulti(:,k)
!               WRITE(*,*) 'Phos:',PhosMulti(:,k)
!       END DO
!       DO k = 1,NumPhosStates
!               READ(4,*) DephosMulti(:,k)
!       END DO
        CLOSE(4)

        ! Read in Simulation input, initial receptor locations, & domain locations
```

```
        OPEN (1,file='BMIP')
        OPEN (2,file='InitialParticleLoc')
        OPEN (7,file='DomainVertices_1')
        OPEN (3,file='DomainLimits_1')
        OPEN (8,file='DomainVertices_2')
        OPEN (4,file='DomainLimits_2')

        DomainLimitsFile(1) = 3
        DomainLimitsFile(2) = 4
        DomainVertFile(1) = 7
        DomainVertFile(2) = 8

        ALLOCATE (Simulation%NumberDomains(NumMonomerSpecies))

        WRITE(*,*) 'Importing Input Data'

        ! Read in values from input file
        READ(1,107) Simulation%OutputDirectory ! HPC Path info
        READ(1,100) Simulation%NumberReceptors ! # of particles
        READ(1,101) Simulation%TimeStep ! Time step [s]
        READ(1,102)
Membrane%xlimmax,Membrane%ylimmax,Membrane%xlimmin,Membrane%ylimmin !
simulation boundaries
        READ(1,*) Simulation%FrameRate ! data print frequency
        READ(1,106) Simulation%Length
        READ(1,109) Simulation%NumberDomains(:) ! simulation length [s], # of
domains [ErbB2 ErbB3]

        ALLOCATE
(Receptor(Simulation%NumberReceptors),ReceptorData(Simulation%NumberReceptors
))
        ALLOCATE (Species(NumBaseSpecies))
        ALLOCATE
(Simulation%NumberVertices(Simulation%NumberDomains(2),NumMonomerSpecies))


        DO k = 1,NumBaseSpecies
                Species(k)%Liganded(:)=0
                Species(k)%Phosphorylated(:)=0
        END DO

        WRITE(*,*) 'Importing Individual Receptor Data'

        ! Read intial position of Receptor
        DO k = 1,Simulation%NumberReceptors
```

```
                READ(2,103)
Receptor(k)%InitialPosition(1),Receptor(k)%InitialPosition(2), &
                                        &Receptor(k)%Domain,
Receptor(k)%OriginalSpecies, Receptor(k)%Ligand, &
                                        &Receptor(k)%BoundBuddy
                103 FORMAT(F18.16,F18.16,I3,I2,I2,I4)

        SpeciesCount(Receptor(k)%OriginalSpecies)=SpeciesCount(Receptor(k)%Origin
alSpecies)+1

        Species(Receptor(k)%OriginalSpecies)%Liganded(Receptor(k)%Ligand+1)= &

        &Species(Receptor(k)%OriginalSpecies)%Liganded(Receptor(k)%Ligand+1)+1
        END DO
        CLOSE(2)

        !WRITE(*,*) Receptor(:)%Domain

        Receptor(:)%Species = Receptor(:)%OriginalSpecies
        Receptor(:)%Position(1) = Receptor(:)%InitialPosition(1)
        Receptor(:)%Position(2) = Receptor(:)%InitialPosition(2)

100 FORMAT(I10)
101 FORMAT(F10.7)
102 FORMAT(F7.4,F7.4,F7.4,F7.4)
105 FORMAT(I10)
106 FORMAT(F6.2)
107 FORMAT(a)
109 FORMAT(I3,I3)
        CLOSE(1)

        WRITE(*,*) 'Importing Domain Data'

        ALLOCATE (Domain(Simulation%NumberDomains(2),NumMonomerSpecies))
        ALLOCATE
(Domain(Simulation%NumberDomains(2),NumMonomerSpecies)%xmin)
        ALLOCATE
(Domain(Simulation%NumberDomains(2),NumMonomerSpecies)%xmax)
        ALLOCATE
(Domain(Simulation%NumberDomains(2),NumMonomerSpecies)%ymin)
        ALLOCATE
(Domain(Simulation%NumberDomains(2),NumMonomerSpecies)%ymax)

        ! Read in number of vertices for each dom
        READ(DomainVertFile(1),*)
Simulation%NumberVertices(1:Simulation%NumberDomains(1),1)
```

```
        READ(DomainVertFile(2),*) Simulation%NumberVertices(:,2)

        DO i = 1,2
                DO j = 1,Simulation%NumberDomains(i)

                        ! Import Domain mins and maxes
                        ALLOCATE (Domain(j,i)%xmin)
                        ALLOCATE (Domain(j,i)%xmax)
                        ALLOCATE (Domain(j,i)%ymin)
                        ALLOCATE (Domain(j,i)%ymax)
                        READ(DomainLimitsFile(i),*)                Domain(j,i)%xmin,
Domain(j,i)%xmax, Domain(j,i)%ymin, Domain(j,i)%ymax

                        ! Import Domain Vertices
                        ALLOCATE
(Domain(j,i)%XCoord(Simulation%NumberVertices(j,i)))
                        ALLOCATE
(Domain(j,i)%YCoord(Simulation%NumberVertices(j,i)))

                        READ(DomainVertFile(i),*) Domain(j,i)%XCoord
                        READ(DomainVertFile(i),*) Domain(j,i)%YCoord

                END DO
        END DO

        CLOSE(3)
        CLOSE(4)
        CLOSE(7)
        CLOSE(8)

        WRITE(*,*) 'Initializing Files'

        ! Create output files

        WRITE(fnstring3,9000) TRIM(Simulation%OutputDirectory)
        WRITE(fnstring4,9001) TRIM(Simulation%OutputDirectory)
        WRITE(fnstring5,9002) TRIM(Simulation%OutputDirectory)
        WRITE(fnstring6,9003) TRIM(Simulation%OutputDirectory)
        WRITE(fnstring7,9004) TRIM(Simulation%OutputDirectory)

9000 FORMAT(a,'/TrueDimerLifetimes')
9001 FORMAT(a,'/MSDData')
9002 FORMAT(a,'/PhosReactions')
9003 FORMAT(a,'/PhosLifetimes')
9004 FORMAT(a,'/DomainExitInfo')
```

```
        OPEN(4,file=fnstring3) ! dimer lifetimes from the actual simulation, not the frame rate
        OPEN(7,file=fnstring4) ! MSD info written to according to frame rate, calculated each dt
        OPEN(8,file=fnstring5) ! time to phosphorylation for each dimer
        OPEN(9,file=fnstring6) ! Phosphorylation time
        OPEN(10,file=fnstring7) ! Exit rate info for Adam

        ! Initialize variables
        Simulation%CurrentTimeStep = 0
        Receptor(:)%Phosphate = 0
        ReceptorData(:)%DimerStart = 0
        ReceptorData(:)%PhosStart = 0
        ReceptorData(:)%JumpSize(1) = 0
        ReceptorData(:)%JumpSize(2) = 0
        ReceptorData(:)%JumpSize(3) = 0
        Receptor(:)%ActiveTail = 0
        ReceptorData(:)%PhosSuccess = 0

        ! Account for double picking of a dimer on rates **Need to make general**
        DimOffRate((NumMonomerSpecies+1):NumBaseSpecies,:)=DimOffRate((NumMonomerSpecies+1):NumBaseSpecies,:)/2
        PhosRate((NumMonomerSpecies+1):NumBaseSpecies,:)=PhosRate((NumMonomerSpecies+1):NumBaseSpecies,:)/2
        DePhosRate((NumMonomerSpecies+1):NumBaseSpecies,:)=DePhosRate((NumMonomerSpecies+1):NumBaseSpecies,:)/2

        ! Initialize Random Seed and Seed Random Number Generator
        CALL SYSTEM_CLOCK(COUNT=seed)
        ! Seed grnd()
        CALL sgrnd(seed)
        ! Calculate number of moves
        NMoves = Simulation%Length/Simulation%TimeStep
        ! Divid number of moves among all receptors
        RMoves = NMoves*Simulation%NumberReceptors
        ! Calculate time step per Receptor
        Simulation%RTimeStep=Simulation%TimeStep/Simulation%NumberReceptors
        ! Calculate datacut frequency
        frames = INT(Simulation%FrameRate*Simulation%Length)
        datacut = INT((RMoves)/(Simulation%FrameRate*Simulation%Length))
        framecount = 0
        datacutcount=0
        WRITE(*,*) 'Main Loop Started'
        ! MAIN SIMULATION LOOP
        DO j = 1,RMoves
```

```
! Update time
Simulation%CurrentTimeStep=Simulation%CurrentTimeStep + Simulation%RTimeStep
! Pick current receptor
i=1+INT((Simulation%NumberReceptors-1)*grnd())
! initialize probabilites
PhosProb = 0
UnbindProb = 0
DephosProb = 0
! Move receptor
CALL ReceptorDiffuse(i,Reaction)

IF (.NOT. Reaction) THEN ! Check for other possible Reactions (Dissociation, Phosphorylation, Dephosphorylation)
!WRITE(*,*) 'Check for other reaction'
IF (Receptor(i)%BoundBuddy > 0) THEN ! Dimer
LigandCount = Receptor(i)%Ligand + Receptor(Receptor(i)%BoundBuddy)%Ligand + 1
PhosphateCount = Receptor(i)%Phosphate + Receptor(Receptor(i)%BoundBuddy)%Phosphate + 1
ELSE ! Monomer
LigandCount = Receptor(i)%Ligand + 1
PhosphateCount = Receptor(i)%Phosphate + 1
END IF
! Calculate Unbinding Probability f(species, ligand)
UnbindProb = DimOffRate(Receptor(i)%Species,LigandCount)*Simulation%TimeStep
!WRITE(*,*) 'Current Species =', Receptor(i)%Species, 'Original Species =', Receptor(i)%OriginalSpecies
!WRITE(*,*) 'LigandCount =', LigandCount, 'PhosphateCount', PhosphateCount
!WRITE(*,*) 'Active Tail =', Receptor(i)%ActiveTail
! Calculate Phosphorylation Probability f(species, receiver phosphorylation state)
IF (Receptor(i)%ActiveTail > 0) THEN ! Dimer, ActiveTail is the receptor in the dimer that is active (to be phosphorylated)
!WRITE(*,*) 'Active Tail Species =', Receptor(Receptor(i)%ActiveTail)%OriginalSpecies
!WRITE(*,*) 'Receiver Phosphorylation State =', (Receptor(Receptor(Receptor(i)%ActiveTail)%BoundBuddy)%Phosphate + 1)
!PhosProb = PhosMulti(Receptor(Receptor(i)%ActiveTail)%OriginalSpecies,PhosphateCount) &
!
&*PhosRate(Receptor(i)%Species,PhosphateCount)*Simulation%TimeStep
```

```
        PhosProb=PhosRate(Receptor(Receptor(Receptor(i)%ActiveTail)%BoundBuddy)
%OriginalSpecies,&

        &(Receptor(Receptor(Receptor(i)%ActiveTail)%BoundBuddy)%Phosphate      +
1))*Simulation%TimeStep
                        ELSE
!                               PhosProb                                        =
PhosMulti(Receptor(i)%OriginalSpecies,PhosphateCount) &
!
        &*PhosRate(Receptor(i)%Species,PhosphateCount)*Simulation%TimeStep

        !PhosProb=PhosRate(Receptor(Receptor(i)%BoundBuddy)%OriginalSpecies,&
                                !
        &(Receptor(Receptor(i)%BoundBuddy)%Phosphate + 1))*Simulation%TimeStep
                                PhosProb = 0
                        END   IF
                        ! Avoid double phosphorylation in a single lifetime
                        IF (ReceptorData(i)%PhosSuccess == 1) THEN
                                PhosProb = 0
                        END IF

                        ! Calculate Dephosphorylation Probability f(species)
                        !DephosProb                                             =
DephosMulti(Receptor(i)%OriginalSpecies,PhosphateCount) &
                        !
        &*DePhosRate(Receptor(i)%Species,PhosphateCount)*Simulation%TimeStep
                        IF (Receptor(i)%Phosphate == 1) THEN

        DephosProb=DePhosRate(Receptor(i)%OriginalSpecies,PhosphateCount)*Simul
ation%TimeStep
                        END IF

                        !IF (Receptor(i)%Species == 4) THEN
!                               WRITE(*,*)              'Species              =              ',
Receptor(i)%Species,'UnbindProb =', UnbindProb, 'PhosProb =', PhosProb, 'DephosProb
=', DephosProb
!                               WRITE(*,*) 'Ligand Count =', LigandCount
!                               WRITE(*,*)      'PhosState      of      ActiveTail      =',
Receptor(Receptor(i)%ActiveTail)%Phosphate
!                       END IF
                        ! Calculate total probability
                        !ProbTot=UnbindProb+PhosProb+DephosProb
                        ! Normalize probabilities
                        !UnbindProbNorm=UnbindProb/ProbTot
                        !PhosProbNorm=PhosProb/ProbTot
```

```
                    !DesphosProbNorm=DephosProb/ProbTot
                    ! Find reaction based on probability
                    !WRITE(*,*) 'Ceiling: ',
CEILING(UnbindProb+PhosProb+DephosProb), 'Sum: ',
UnbindProb+PhosProb+DephosProb
                    IF (CEILING(UnbindProb+PhosProb+DephosProb)>0) THEN
                        ! Generate random number
                        randnum=grnd()
                        IF (randnum > 0 .AND.randnum <= PhosProb) THEN
                            ! Phosphorylate
                            !IF (Receptor(i)%Species == 4) THEN
!                               WRITE(*,*) 'Out of Function:
PHOSPHORYLATE! Species: ', Receptor(i)%Species!, 'Ligand Count: ', LigandCount,
'Phos Count: ', PhosphateCount
!                               !WRITE(*,*) 'Species = ',
Receptor(i)%Species,'UnbindProb =', UnbindProb, 'PhosProb =', PhosProb, 'DephosProb
=', DephosProb
!                               WRITE(*,*) 'Ligand Count (+1 for
indexing)=', Receptor(i)%Ligand + Receptor(Receptor(i)%BoundBuddy)%Ligand + 1
!                               WRITE(*,*) 'PhosState of ActiveTail =',
Receptor(Receptor(i)%ActiveTail)%Phosphate
!                           END IF
                            !Write(*,*) 'Min Prob: 0, Max Prob: ', PhosProb
                            !WRITE(*,*) 'Unbind: ', UnbindProb, '**Phos: ',
PhosProb, 'Dephos: ', DephosProb, 'Rand #: ', randnum


                            IF (Receptor(Receptor(i)%ActiveTail)%Phosphate
== 0) THEN
                                CALL Phosphorylate(i)
                            END IF
                        ELSEIF (randnum > PhosProb .AND.randnum <=
UnbindProb+PhosProb) THEN
                            ! Unbind
                            !WRITE(*,*) 'UNBIND! Species: ',
Receptor(i)%Species, 'Ligand Count: ', LigandCount, 'Phos Count: ', PhosphateCount
                            !Write(*,*) 'Min Prob: ', PhosProb, ' Max Prob:
',UnbindProb+PhosProb
                            !WRITE(*,*) '**Unbind: ', UnbindProb, 'Phos: ',
PhosProb, 'Dephos: ', DephosProb, 'Rand #: ', randnum
                            CALL DissociationReaction(i)
                        ELSEIF (randnum > UnbindProb+PhosProb
.AND.randnum <= UnbindProb+PhosProb+DephosProb) THEN
                            ! Dephosphorylate
                            !WRITE(*,*) 'DEPHOSPHORYLATE! Species: ',
Receptor(i)%Species, 'Ligand Count: ', LigandCount, 'Phos Count: ', PhosphateCount
```

```
                                        !Write(*,*) 'Min Prob: ', UnbindProb+PhosProb, '
Max Prob: ', UnbindProb+PhosProb+DephosProb
                                        !WRITE(*,*) 'Unbind: ', UnbindProb, 'Phos: ',
PhosProb, '**Dephos: ', DephosProb, 'Rand #: ', randnum
                                        CALL Dephosphorylate(i)
                                    END IF
                                END IF
                            END IF
                            ! Record keeping code
                            datacutcount=datacutcount+1

                            !WRITE(*,*) 'DataCut:', datacutcount, '/', datacut
                            !WRITE(*,*) 'FrameCount:', framecount

                            !!!!!
                            ! Check if the data should be written to a file
                            IF (datacutcount == datacut) THEN
                                framecount=framecount+1
                                ! generate filename for data storage
                                IF (framecount < 10) THEN
                                    WRITE(fnstring,                                    7001)
TRIM(Simulation%OutputDirectory), framecount

                                ELSE IF (framecount < 100) THEN
                                    WRITE(fnstring,                                    7002)
TRIM(Simulation%OutputDirectory), framecount

                                ELSE IF (framecount < 1000) THEN
                                    WRITE(fnstring,                                    7003)
TRIM(Simulation%OutputDirectory), framecount

                                ELSE IF (framecount < 10000) THEN
                                    WRITE(fnstring,                                    7004)
TRIM(Simulation%OutputDirectory), framecount

                                ELSE IF (framecount < 100000) THEN
                                    WRITE(fnstring,                                    7005)
TRIM(Simulation%OutputDirectory), framecount

                                ELSE
                                    WRITE(fnstring,                                    7006)
TRIM(Simulation%OutputDirectory), framecount

                                END IF

                                OPEN(77,file=fnstring)
```

```
                    !WRITE(*,*) 'New Data File'
                    WRITE(77,*) 'Rows: Receptor Columns: Receptor #, x, y, dx, dy,
d, monomer/dimer, &
                    &ligand, Bound Particle #, Domain, Phosphorylation state'

                    ! Write data to file
                    DO i = 1,Simulation%NumberReceptors
                        IF (Receptor(i)%Species <= NumMonomerSpecies) THEN
                            WRITE(77,*)  i,  Receptor(i)%Position(1),
Receptor(i)%Position(2), ReceptorData(i)%JumpSize(1), &
                                &ReceptorData(i)%JumpSize(2),
ReceptorData(i)%JumpSize(3), 1, Receptor(i)%Ligand, &
                                &Receptor(i)%BoundBuddy,
Receptor(i)%Domain, Receptor(i)%Phosphate
                        ELSE
                            WRITE(77,*)  i,  Receptor(i)%Position(1),
Receptor(i)%Position(2), ReceptorData(i)%JumpSize(1), &
                                &ReceptorData(i)%JumpSize(2),
ReceptorData(i)%JumpSize(3), 2, Receptor(i)%Ligand, &
                                &Receptor(i)%BoundBuddy,
Receptor(i)%Domain, Receptor(i)%Phosphate
                        END IF
                    END DO

                    CLOSE(77)

                    datacutcount=0
                    !WRITE(*,*) 'Species Count: ', SpeciesCount(:)
                END IF

7001                FORMAT(a,'/Data_Files/ParticleData.',I1);
7002                FORMAT(a,'/Data_Files/ParticleData.',I2);
7003                FORMAT(a,'/Data_Files/ParticleData.',I3);
7004                FORMAT(a,'/Data_Files/ParticleData.',I4);
7005                FORMAT(a,'/Data_Files/ParticleData.',I5);
7006                FORMAT(a,'/Data_Files/ParticleData.',I6);


        END DO
        CLOSE(4)
        CLOSE(8)
        CLOSE(9)
        WRITE(*,*) 'Simulation Completed'

END PROGRAM ErbB23_Sim
```

APPENDIX C: Chapter 3 Supplement

The foundation of this model is a Brownian motion simulator with three separate modules (Figure A.1A). The first module performs initial data processing in Matlab using graphical user interface (GUI, Figure A.1B). This function facilitates design of the membrane simulation space based upon an EM image, providing the number of receptors, their distribution on the membrane, the estimated area and position of confinement zones. Module 2 is executed in FORTRAN for Brownian motion simulation, dimerization and phosphorylation; the reactions are governed by modified Smoluchowski kinetics. Module 3 performs post data processing in Matlab.

Particle Diffusion. Particle diffusion is based on Brownian motion. Brownian motion is represented in a simulation by picking a random number from a normal distribution and applying that value to the root mean square (RMS) step length, creating the stochastic nature of diffusion (Andrews & Bray, 2004; Kusumi et al, 1993; Popov & Agmon, 2001).

$$x(t+\Delta t) = x(t) + RMS\xi_x$$
$$y(t+\Delta t) = y(t) + RMS\xi_y$$
$$RMS = \sqrt{2D\Delta t}$$

where x and y are the particle's Cartesian coordinates, RMS is the root mean square step, Dt is the time step, and ξ_x and ξ_y are the normally distributed random numbers. Periodic boundary conditions are used for the boundaries of the simulation space. Confinement zones, as imported from the EM image, create a hurdle for the diffusing particles during the simulation. Particles are free to diffuse into confinement zones,

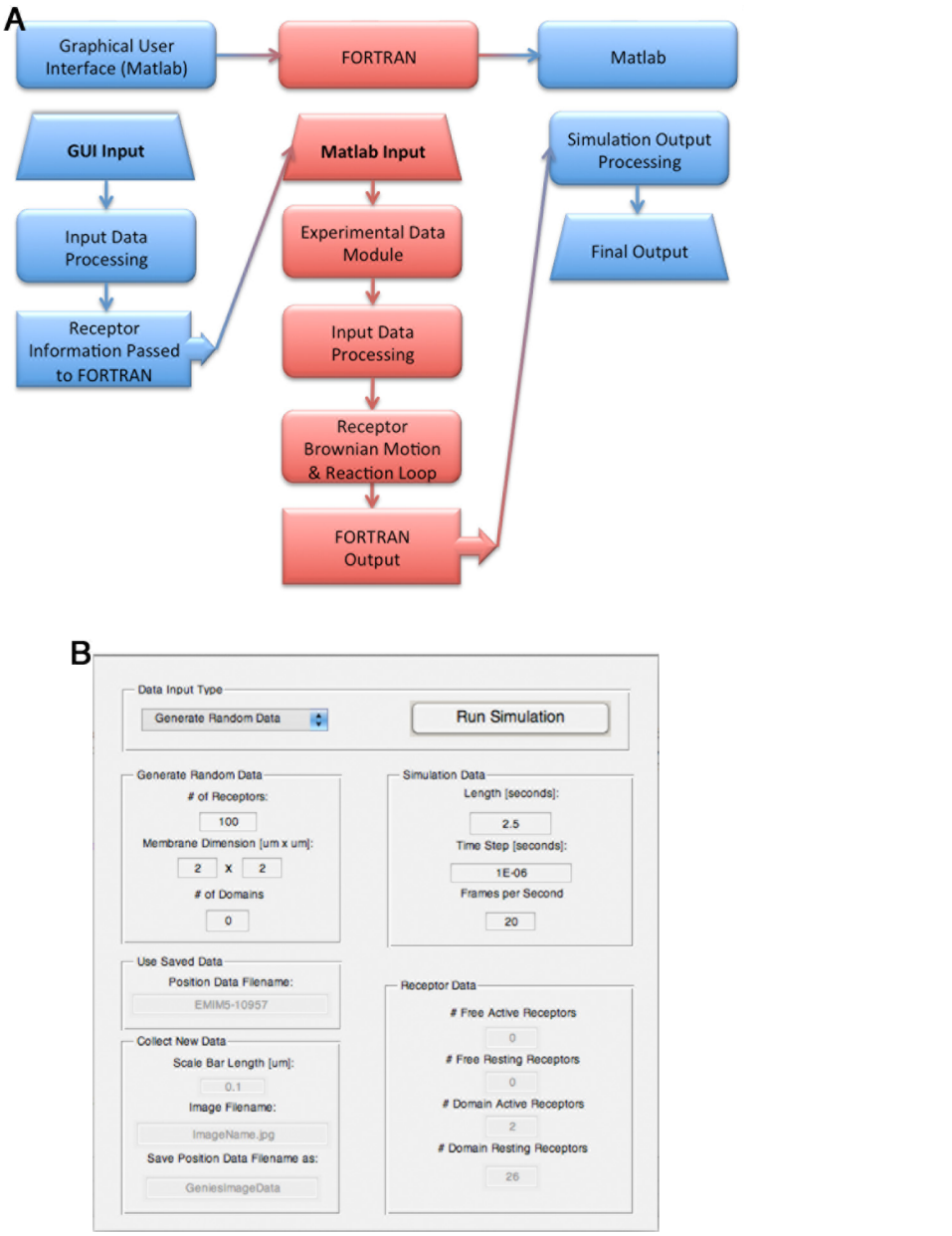

Figure A.1: Model workflow and user interface. (A) Model workflow from user initiation through final output. Initialization of the simulation occurs through a graphical user interface (GUI) allowing the user to vary different simulation conditions. Input data collected through the GUI is then passed through to the main simulator, which executes the desired simulation. Finally, the simulation output is passed to post processing scripts to run relevant analysis and generate plots. (B) Graphical user interface used to pass through variable model inputs to control simulation conditions.

however receptors must pay a "toll" to escape the confinement zone. A probability is calculated using escape rates calculated from the Single Particle Tracking (SPT) experiments and the time step, similar to the previous version of the model by Hsieh et al. (Hsieh et al, 2008):

However, this probability has been simplified further to:

due to the small time step. If a confined particle is set to diffuse out of a confinement zone, and this escape probability is not met, the confinement zone assumes a reflective boundary condition, trapping the particle inside.

Kinetics. Modified Smoluchowski kinetics are used to simulate particle kinetics (see Andrews and Bray (Andrews & Bray, 2004)). During simulation, rules for dimerization and phosphorylation determine the course of the receptor in question. The simulator focuses on one receptor at a time, picking every receptor once over a fixed time step. Receptors are picked randomly, allowing the order the receptors are moved or reacted to be different for each time step. Once a receptor is picked, a series of reactions are considered before the fate of the receptor is implemented. There are four possible reactions that can take place, dimerization, dimer dissociation, phosphorylation, and dephosphorylation. Dimerization is treated as a second-order reaction, while dimer dissociation, phosphorylation, and dephosphorylation are all treated as first-order reactions.

Dimerization

A receptor's likelihood to dimerize with another receptor is based on a distance termed the binding radius. The binding radius takes into account the dimer on rate, diffusion coefficient of the receptors that will comprise the dimer, and the simulation time step (Andrews & Bray, 2004). While the binding radius is not a physical radius, the use of this radius allows the simulation to be closer to the physical situation than previous methods using probabilities (Andrews & Bray, 2004; Erban & Chapman, 2009; Gillespie, 1977; Hsieh et al, 2008; Popov & Agmon, 2001). When the chosen receptor is moved, the final position is scanned for other receptors that are within the binding radius. If a receptor is within this distance, then the receptors will form a dimer. Determination of the binding radius is discussed in model parameters below.

Dimer Dissociation, Phosphorylation, & Dephosphorylation

Dimer dissociation and dephosphorylation are first order reactions. Dimer phosphorylation may be considered second-order due to the dimer and phosphate reacting, however here phosphate is assumed to be in excess, therefore the reaction can be approximated as first-order. First-order reactions are implemented through a probability calculated using the reaction rate and simulation time step:

This probability has been simplified further, as in Hsieh et al (Hsieh et al, 2008) and Andrews and Bray (Andrews & Bray, 2004):

When dimer dissociation occurs, an unbinding radius is used to set the dissociating receptors apart. The unbinding radius is calculated such that the occurrence of an unrealistic amount of repeated interactions is minimized:

$$\phi_s = \frac{\sigma_b}{\sigma_u}$$

where σ_b is the binding radius and σ_u is the unbinding radius. The default ratio of binding radius to unbinding radius is 0.2.

Model Parameters. The majority of parameters needed for the model are directly measured using SPT experiments (Low-Nam et al, 2011). These parameters include dimer off rates, diffusion coefficients, and confinement zone escape rates. Phosphorylation and dephosphorylation rates were measured by Kleiman et al (Kleiman et al, 2011). The most elusive model parameter is the dimer on rate, which cannot currently be accurately measured through SPT. Martin-Fernandez (Martin-Fernandez et al, 2002) found 14% of receptors on a membrane participate in preformed dimers at steady state. Using this information, a binding radius is calculated by iterating the binding radius over a large range until the percent of preformed dimers converged to approximately 14% during 4 minute simulations. This binding radius is then used to back calculate the dimer on rate using SMOLDYN (Andrews & Bray, 2004).

Model Validation. Parameters were validated by comparison to experimental data. This is shown in Figure 3.1EF, where simulated diffusion coefficients and off rates match well with the experimental values in Low-Nam et al. (Low-Nam et al, 2011). Plots in Figure A.2A show good comparison of phosphorylation kinetics in the model versus experimentally measured values in MCF7 cells (Verveer et al, 2000). In addition,

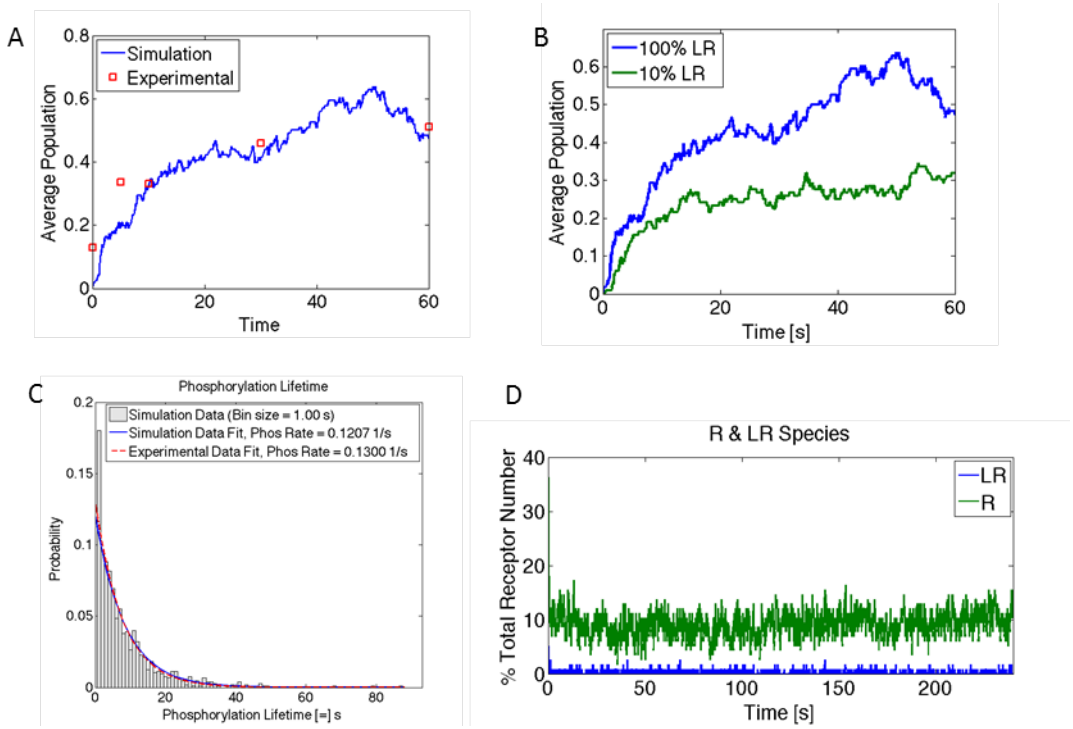

Figure A.2: ErbB1 Receptor Phosphorylation (A) Average phosphorylation population over a 60 second simulation as compared to experimental data from Verveer et al. (Verveer et al, 2000) (B) Average phosphorylation population for a 10% LR simulation and 100% LR simulation. (C) Comparison of the simulated phosphorylation lifetime in simulation against experimental values reported by Kleiman et al. (8). (D) Percent of total monomers relative to ligand occupancy, over a 4 minute simulation with 50% LR.

Figure A.2C below reports model validation for phosphorylation lifetimes, as compared to experimental values reported for dephosphorylation of EGFR after acute exposure to the kinase inhibitor Gefitinib (Kleiman et al, 2011). Figure A.2D shows a shift in available monomer equilibrium for a 50% LR simulation. Initially the simulation has equal amounts of unliganded monomer (R) and liganded monomer (LR) available. As reactions occur and receptors transition to different states, the equilibrium of available monomer transitions from equal amounts, to an excess of R monomers. LRLR dimers form whenever two LR monomers are within the binding radius. LRR dimers have a 0.1% chance of forming within the binding radius, compared to a 0.01% chance for RR dimer, due to the conformational flux of R monomers.

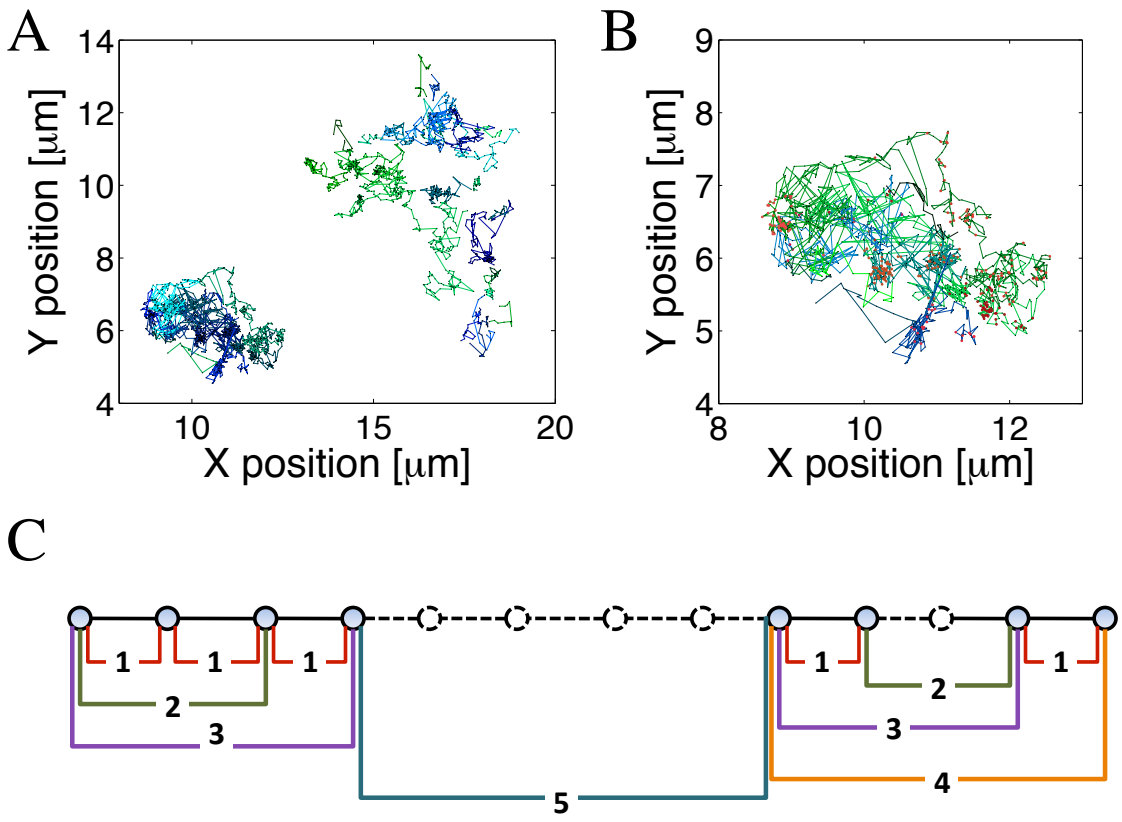

Figure A.3: Details of the experimental data and analysis method. A. Raw trajectories as captured by single particle tracking analysis. Positions are determined by fitting a distribution to the light intensity captured in a set of neighboring pixels. Trajectories in this recording extend over a rectangle of approximately 25μm Å~ 45μm B. Detail of the trajectories shown in A. Red circles label points that are associated with 2-step displacements in the lower 70% of 2-step displacements in the entire recording. The labelled points indicate limited movement. Different trajectories (represented by color) tend to slow down in the same locations. As the trajectories are not simultaneous (due to the imaging method), this may indicate that the regions of slow movement are localized. C. Given an incomplete record of a trajectory, for every integer multiple of the time step, we choose as many non-overlapping pairs of recorded positions as possible.

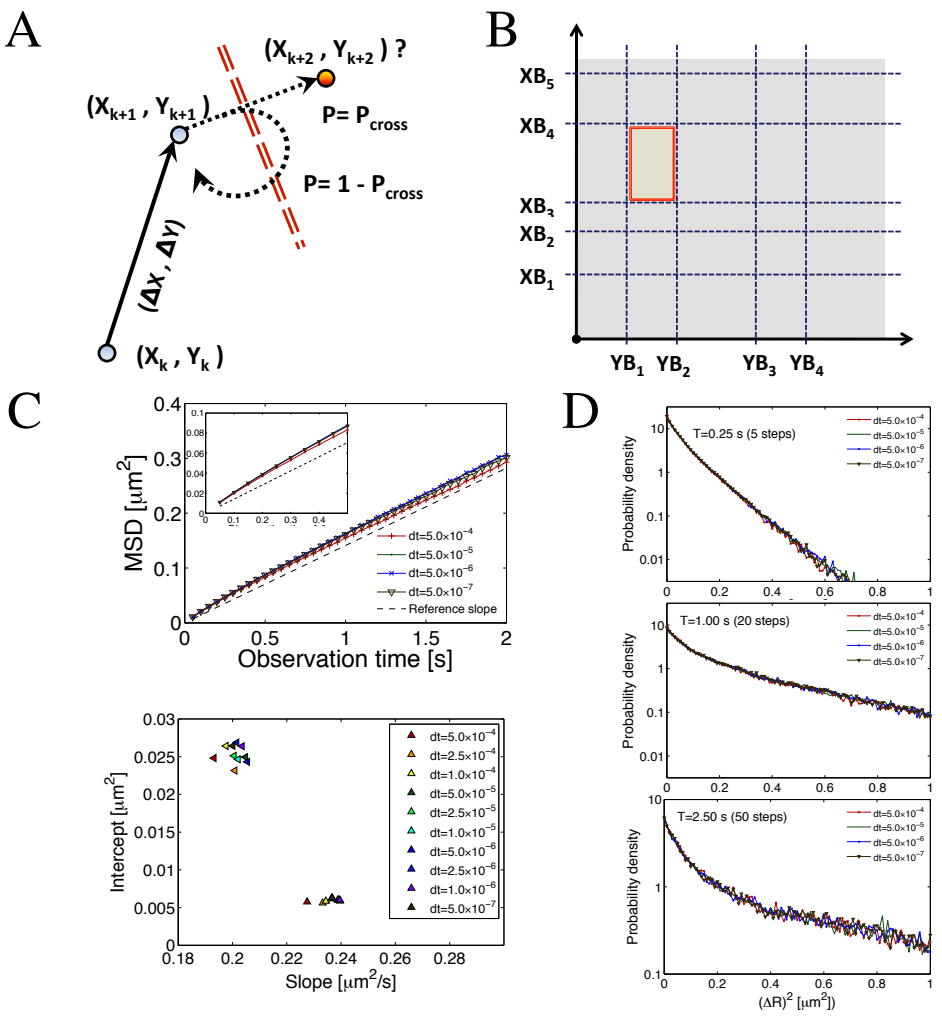

Figure A.4: Details of the numerical simulation. A. At each iteration, a tentative new position is generated for each particle. The new position is accepted if there is no barrier between it and the current position; otherwise it is accepted with probability p_{cross}. In case of rejection, the particle keeps its current position for another time step. B. The simulation landscape consists of unevenly spaced rectangular barriers, induced by two sets of barrier coordinates, $\{X_{B1}, \cdots, X_{BN_B}\}$ and $\{Y_{B1}, \cdots, Y_{BN_B}\}$; domains are rectangles of the form $[X_{Bk}, X_{Bk+1}]$ Å~ $[Y_{Bl}, Y_{Bl+1}]$. Barrier permeabilities may depend on the direction of the attempted crossing. C-D. Sensitivity to the simulation time step t_{sim}. The MSD versus time distribution (C, top) varies only slightly when t_{sim} varies over several orders of magnitude. The bottom panel shows slope-intercept pairs corresponding to the first 10 (observation) time steps t_{obs} (upward pointing triangles, lower group) and steps 20-100 (left pointing triangles, upper group). The fixed time displacement distributions are also robust (D).

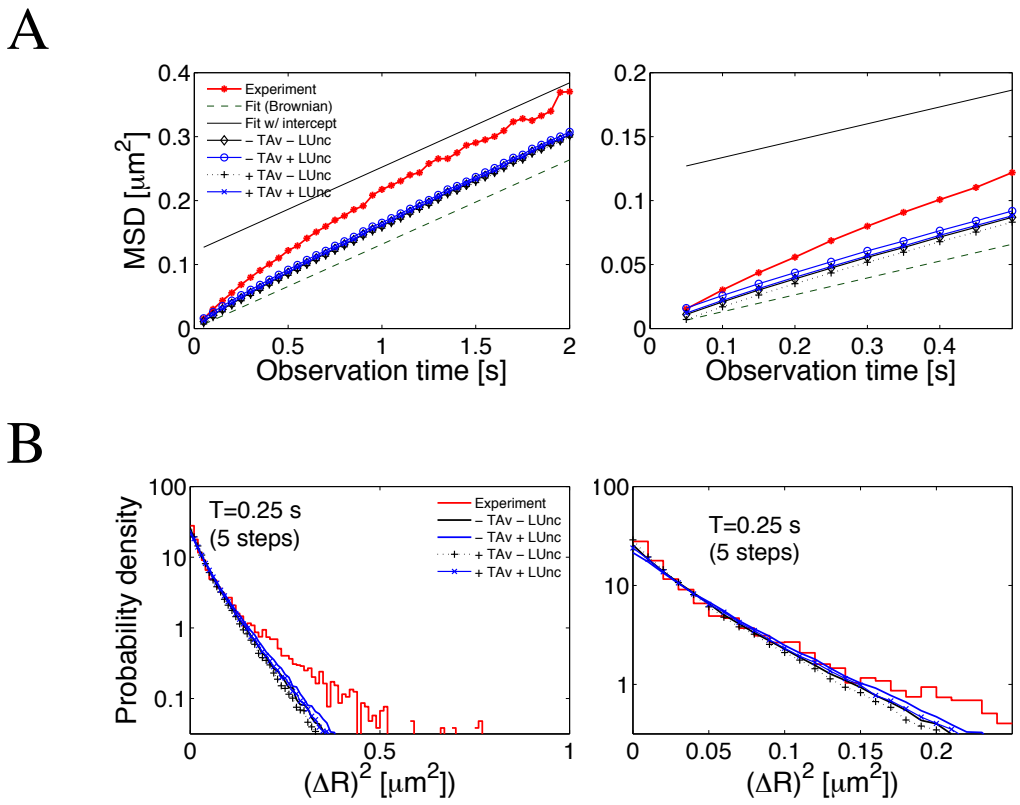

Figure A.5: Corrections for time averaging and localization uncertainty. Simulation results with and without corrections for localization uncertainty and / or recording time averaging (results shown in Fig. 4 have both corrections). A. Mean square displacement (MSD) versus time. The experimental data, a simple Brownian (zero intercept) as well as a linear fit to it are shown for comparison. The right panel shows that same results at a higher resolution. B. Distribution of square displacements after 20 steps, corresponding to 1.0s. In both types of results, time averaging tends to reduce, and localization uncertainty tends to increase the observed square displacements.

APPENDIX E: Chapter 5 Supplement

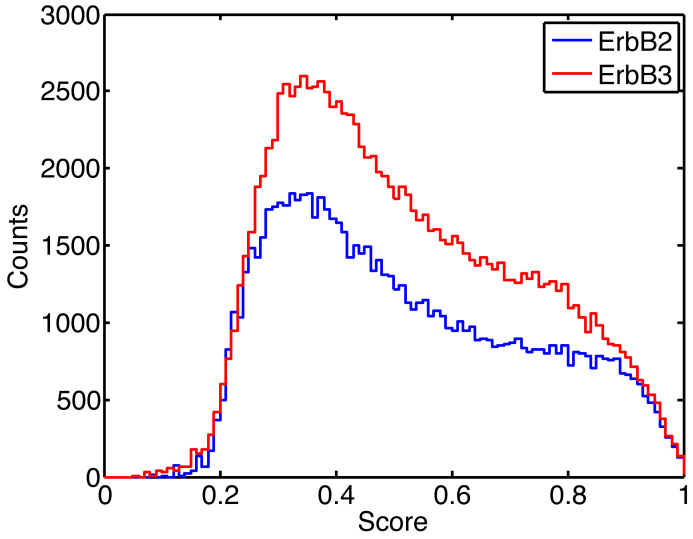

Figure A.6: Compiled ranks for all the points from SPT data for ErbB2 and ErbB3. A biomodal distribution is apparent for both speices. This is a compilation for tracking erbB2 with QD-655 and erbB3 with QD-585.

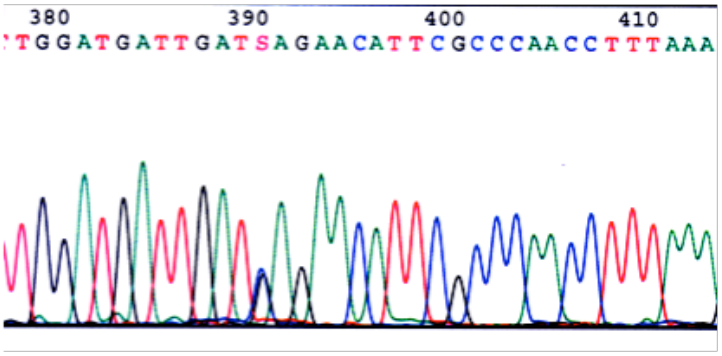

Figure A.7: Amino acid substitution mutation in one of the erbB3 alleles in the widely-used SKBR3 breast cancer cell line.

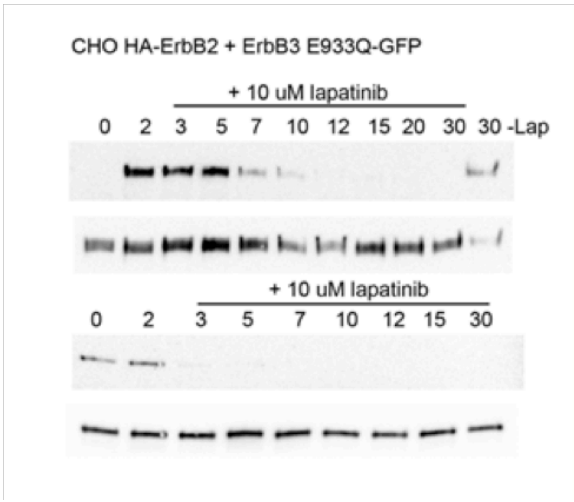

Figure A.8: ErbB3 E933Q Dephosphorylation. An activation/inhibition assay shows high levels of phosphorylated erbB3 up to the 5 minute time point.

REFERENCES

Abulrob A, Lu Z, Baumann E, Vobornik D, Taylor R, Stanimirovic D, Johnston LJ (2010) Nanoscale imaging of epidermal growth factor receptor clustering: effects of inhibitors. *The Journal of biological chemistry* **285:** 3145-3156

Adak S, DeAndrade D, Pike LJ (2011) The tethering arm of the EGF receptor is required for negative cooperativity and signal transduction. *The Journal of biological chemistry* **286:** 1545-1555

Akira S, Takeda K (2004) Toll-like receptor signalling. *Nature reviews Immunology* **4:** 499-511

Alberts B (2008) *Molecular biology of the cell*, 5th edn. New York: Garland Science.

Allen JA, Halverson-Tamboli RA, Rasenick MM (2007) Lipid raft microdomains and neurotransmitter signalling. *Nat Rev Neurosci* **8:** 128-140

Andrews NL, Lidke KA, Pfeiffer JR, Burns AR, Wilson BS, Oliver JM, Lidke DS (2008) Actin restricts FceRI diffusion and facilitates antigen-induced receptor immobilization. *Nat Cell Biol* **10:** 955-963

Andrews SS, Addy NJ, Brent R, Arkin AP (2010) Detailed simulations of cell biology with Smoldyn 2.1. *PLoS Comput Biol* **6:** e1000705

Andrews SS, Bray D (2004) Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Physical Biology* **1:** 137-151

Ariotti N, Liang H, Xu Y, Zhang Y, Yonekubo Y, Inder K, Du G, Parton RG, Hancock JF, Plowman SJ (2010) Epidermal growth factor receptor activation remodels the plasma membrane lipid environment to induce nanocluster formation. *Molecular and cellular biology* **30:** 3795-3804

Auerbach SM (2000) Theory and simulation of jump dynamics, diffusion and phase equilibrium in nanopores. *Int Rev Pnys Chem* **19:** 155-198

Axelrod D, Koppel DE, Elson E, Webb WW (1976) Mobility measurement by analysis of fluorescence photobleaching recovery kinetics. *Biophys J* **16:** 1055-1069

Bader AN, Hofman EG, Voortman J, en Henegouwen PM, Gerritsen HC (2009) Homo-FRET imaging enables quantification of protein cluster sizes with subcellular resolution. *Biophys J* **97:** 2613-2622

Baselga J, Swain SM (2009) Novel anticancer targets: revisiting ERBB2 and discovering ERBB3. *Nature Reviews Cancer* **9:** 463-475

Berger MB, Mendrola JM, Lemmon MA (2004) ErbB3/HER3 does not homodimerize upon neuregulin binding at the cell surface. *FEBS letters* **569:** 332-336

Blinov ML, Faeder JR, Goldstein B, Hlavacek WS (2006) A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity. *Biosystems* **83:** 136-151

Bobroff N (1986) Position measurement with a resolution and noise-limited instrument. *Rev Sci Instrum* **57:** 1152

Brameshuber M, Weghuber J, Ruprecht V, Gombos I, Horvath I, Vigh L, Eckerstorfer P, Kiss E, Stockinger H, Schutz GJ (2010) Imaging of mobile long-lived nanoplatforms in the live cell plasma membrane. *The Journal of biological chemistry* **285:** 41765-41771

Brightman FA, Fell DA (2000) Differential feedback regulation of the MAPK cascade underlies the quantitative differences in EGF and NGF signalling in PC12 cells. *FEBS letters* **482:** 169-174

Brinkerhoff CJ, Woolf PJ, Linderman JJ (2004) Monte Carlo simulations of receptor dynamics: insights into cell signaling. *J Mol Histol* **35:** 667-677

Britten CD (2004) Targeting ErbB receptor signaling: A pan-ErbB approach to cancer. *Mol Cancer Ther* **3:** 1335-1342

Brown GC, Kholodenko BN (1999) Spatial gradients of cellular phospho-proteins. *FEBS letters* **457:** 452-454

Bublil EM, Yarden Y (2007) The EGF receptor family: spearheading a merger of signaling and therapeutics. *Curr Opin Cell Biol* **19:** 124-134

Burgess AW, Cho H-S, Eigenbrot C, Ferguson KM, Garret TPJ, Leahy DJ, Lemmon MA, Silwkowski MX, Ward CW, Yokoyama S (2003) An Open-and-Shut Case? Recent Insights into the Activation of EGF/ErbB Receptors. *Molecular cell* **12:** 541-552

Burrage K, Hancock J, Leier A, Nicolau DV, Jr. (2007) Modelling and simulation techniques for membrane biology. *Brief Bioinform* **8:** 234-244

Cambi A, de Lange F, van Maarseveen NM, Nijhuis M, Joosten B, van Dijk EM, de Bakker BI, Fransen JA, Bovee-Geurts PH, van Leeuwen FN, Van Hulst NF, Figdor CG (2004) Microdomains of the C-type lectin DC-SIGN are portals for virus entry into dendritic cells. *The Journal of cell biology* **164:** 145-155

Casaletto JB, McClatchey AI (2012) Spatial regulation of receptor tyrosine kinases in development and cancer. *Nature reviews Cancer* **12:** 387-400

Chakraborty AK, Dustin ML, Shaw AS (2003) In silico models for cellular and molecular immunology: successes, promises and challenges. *Nat Immunol* **4:** 933-936

Chatterjee A, Mayawala K, Edwards JS, Vlachos DG (2005) Time accelerated Monte Carlo simulations of biological networks using the binomial tau-leap method. *Bioinformatics* **21:** 2136-2137

Cho HS, Leahy DJ (2002) Structure of the extracellular region of HER3 reveals an interdomain tether. *Science* **297:** 1330-1333

Chuan Kang H, Weinberg W (1995) Mondeling the kinetifcs of heterogeneous catalysis. *Chem Rev* **95:** 667-676

Chung I, Akita R, Vandlen R, Toomre D, Schlessinger J, Mellman I (2010) Spatial control of EGF receptor activation by reversible dimerization on living cells. *Nature* **464:** 783-787

Citri A, Yarden Y (2006) EGF-ERBB signalling: towards the systems level. *Nat Rev Mol Cell Biol* **7:** 505-516

Clayton AHA, Tavarnesi ML, Johns TG (2007) Unligated Epidermal Growth Factor Receptor Forms Higher Order Oligomers within Microclusters on A431 Cells That Are Sensitive to Tyrosine Kinase Inhibitor Binding. *Biochemistry* **46:** 4589-4597

Colicelli J (2004) Human RAS superfamily proteins and related GTPases. *Science's STKE : signal transduction knowledge environment* **2004:** RE13

Coppens MO, Bell AT, Chakraborty AK (1999) Dynamic Monte-Carlo and mean-field study of the effect of strong adsorption sites on self-diffusion in zeolites. *Chemical Engineering Science* **54:** 3455-3463

Costa MN, Radhakrishnan K, Edwards JS (2009a) Monte Carlo simulations of plasma membrane corral-induced EGFR clustering. *J Biotechnol* **151:** 261-270

Costa MN, Radhakrishnan K, Wilson BS, Vlachos DG, Edwards JS (2009b) Coupled Stochastic Spatial and Non-Spatial Simulations of ErbB1 Signaling Pathways Demonstrate the Importance of Spatial Organization in Signal Transduction. *Plos One* **4**

Costantino S, Comeau JW, Kolin DL, Wiseman PW (2005) Accuracy and dynamic range of spatial image correlation and cross-correlation spectroscopy. *Biophys J* **89:** 1251-1260

de Keijzer S, Serge A, van Hemert F, Lommerse PHM, Lamers GEM, Spaink HP, Schmidt T, Snaar-Jagalska BE (2008) A spatially restricted increase in receptor mobility is involved in directional sensing during Dictyostelium discoideum chemotaxis. *Journal of cell science* **121:** 1750-1757

Dehmelt L, Bastiaens PI (2010) Spatial organization of intracellular communication: insights from imaging. *Nature reviews Molecular cell biology* **11:** 440-452

Douglass AD, Vale RD (2005) Single-molecule microscopy reveals plasma membrane microdomains created by protein-protein networks that exclude or trap signaling molecules in T cells. *Cell* **121:** 937-950

Edidin M (2001) Lipid microdomains in cell surface membranes. *Current opinion in structural biology* **7:** 528-532

Erban R, Chapman SJ (2009) Stochastic modelling of reaction-diffusion processes: algorithms for bimolecular reactions. *Phys Biol* **6:** 046001

Espinoza FA, Wester MJ, Oliver JM, Wilson BS, Andrews NL, Lidke DS, Steinberg S (2012) Insights into Cell Membrane Microdomain Organization from Live Cell Single Particle Tracking of the IgE High Affinity Receptor FcϵRI of Mast Cells. *Bull Math Biol* **74:** 1857-1911

Faeder J, Blinov M, Hlavacek W (2009) Rules-based modeling of biochemical systems with BioNetGen. *Methods in Molecular Biology* **500:** 113-168

Fallahi-Sichani M, Linderman JJ (2009) Lipid raft-mediated regulation of G-protein coupled receptor signaling by ligands which influence receptor dimerization: a computational study. *PLoS One* **4:** e6604

Feder TJ, Brust-Mascher I, Slattery JP, Baird B, Webb WW (1996) Constrained diffusion or immobile fraction on cell surfaces: a new interpretation. *Biophys J* **70:** 2767-2773

Ferguson KM, Berger MB, Mendrola JM, Cho HS, Leahy DJ, Lemmon MA (2003) EGF activates its receptor by removing interactions that autoinhibit ectodomain dimerization. *Molecular cell* **11:** 507-517

Ferguson KM, Darling PJ, Mohan MJ, Macatee TL, Lemmon MA (2000) Extracellular domains drive homo- but not hetero-dimerization of erbB receptors. *The EMBO journal* **19:** 4632-4643

Forney JE (1973) Who should regulate whom? *Health laboratory science* **10:** 277-279

Friday BB, Adjei AA (2008) Advances in targeting the Ras/Raf/MEK/Erk mitogen-activated protein kinase cascade with MEK inhibitors for cancer therapy. *Clinical cancer research : an official journal of the American Association for Cancer Research* **14:** 342-346

Fujioka A, Terai K, Itoh RE, Aoki K, Nakamura T, Kuroda S, Nishida E, Matsuda M (2006) Dynamics of the Ras/ERK MAPK cascade as monitored by fluorescent probes. *The Journal of biological chemistry* **281:** 8917-8926

Fuxe K, Kenakin T (2010) The changing world of G protein-coupled receptors. *J Recept Signal Transduct Res* **30:** 271

Garrett TP, McKern NM, Lou M, Elleman TC, Adams TE, Lovrecz GO, Kofler M, Jorissen RN, Nice EC, Burgess AW, Ward CW (2003) The crystal structure of a truncated ErbB2 ectodomain reveals an active conformation, poised to interact with other ErbB receptors. *Molecular cell* **11:** 495-505

Gelles J, Schnapp BJ, Sheetz MP (1988) Tracking kinesin-driven movements with nanometre-scale precision. *Nature* **331:** 450-453

Gheber LA, Edidin M (1999) A Model for Membrane Patchiness: Lateral Diffusion in the Presence of Barriers and Vesicle Traffic. *Biophys J* **77:** 3163-3175

Gibson MA, Bruck J (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. *J of Physical Chemistry* **104:** 1876-1889

Gillespie DT (1977) Exact Stochastic Simulation of Coupled Chemical Reactions. *J Phys Chem* **81:** 2340-2361

Gillespie DT (2007) Stochastic simulation of chemical kinetics. *Annu Rev Phys Chem* **58:** 35-55

Gilmer G (1980) Computer models of crystal growth. *Science* **208:** 355-363

Govindan R (2010) A review of epidermal growth factor receptor/HER2 inhibitors in the treatment of patients with non-small-cell lung cancer. *Clin Lung Cancer* **11:** 8-12

Grima R, Schnell S (2008) Modelling reaction kinetics inside cells. *Essays Biochem* **45:** 41-56

Harding AS, Hancock JF (2008) Using plasma membrane nanoclusters to build better signaling circuits. *Trends in cell biology* **18:** 364-371

Hartman NC, Groves JT (2011) Signaling clusters in the cell membrane. *Curr Opin Cell Biol* **23:** 370-376

Hatakeyama M, Kimura S, Naka T, Kawasaki T, Yumoto N, Ichikawa M, Kim JH, Saito K, Saeki M, Shirouzu M, Yokoyama S, Konagaya A (2003) A computational model on the modulation of mitogen-activated protein kinase (MAPK) and Akt pathways in heregulin-induced ErbB signalling. *The Biochemical journal* **373:** 451-463

Heinemann F, Vogel SK, Schwille P (2013) Lateral Membrane Diffusion Modulated by a Minimal Actin Cortex. *Biophys J* **104:** 1465-1475

Hendriks BS, Opresko LK, Wiley HS, Lauffenburger D (2003) Quantitative analysis of HER2-mediated effects on HER2 and epidermal growth factor receptor endocytosis - Distribution of homo- and heterodimers depends on relative HER2 levels. *J Biol Chem* **278:** 23343-23351

Hlavacek W, Faeder J, Blinov M, Posner R, Hucka M, Fontana W (2006) Rules for modeling signal-transduction systems. *Sci STKE* **re6**

Hornberg JJ, Binder B, Bruggeman FJ, Schoeberl B, Heinrich R, Westerhoff HV (2005) Control of MAPK signalling: from complexity to what really matters. *Oncogene* **24:** 5533-5542

Hsieh MY, Yang S, Raymond-Stinz MA, Edwards JS, Wilson BS (2010) Spatio-temporal modeling of signaling protein recruitment to EGFR. *BMC systems biology* **4:** 57

Hsieh MY, Yang S, Raymond-Stinz MA, Steinberg S, Vlachos DG, Shu W, Wilson B, Edwards JS (2008) Stochastic simulations of ErbB homo and heterodimerisation: potential impacts of receptor conformational state and spatial segregation. *IET Syst Biol* **2:** 256-272

Huang S-M, Li C, Armstrong EA, Peet CR, Saker J, Amler LC, Sliwkowski MX, Harari P (2012) Dual targeting of EGFR and HER3 with MEHD7945A overcomes acquired resistance to EGFR inhibitors and radiation. *Cancer Research*

Hynes NE, MacDonald G (2009) ErbB receptors and signaling pathways in cancer. *Curr Opin Cell Biol* **21:** 177-184

Insel PA, Tang CM, Hahntow I, Michel MC (2007) Impact of GPCRs in clinical medicine: monogenic diseases, genetic variants and drug targets. *Biochim Biophys Acta* **1768:** 994-1005

Itano MS, Steinhauer C, Schmied JJ, Forthmann C, Liu P, Neumann AK, Thompson NL, Tinnefeld P, Jacobson K (2012) Super-resolution imaging of C-type lectin and influenza hemagglutinin nanodomains on plasma membranes using blink microscopy. *Biophys J* **102:** 1534-1542

Jaiswal Bijay S, Kljavin Noelyn M, Stawiski Eric W, Chan E, Parikh C, Durinck S, Chaudhuri S, Pujara K, Guillory J, Edgar Kyle A, Janakiraman V, Scholz R-P, Bowman Krista K, Lorenzo M, Li H, Wu J, Yuan W, Peters Brock A, Kan Z, Stinson J, Mak M, Modrusan Z, Eigenbrot C, Firestein R, Stern Howard M, Rajalingam K, Schaefer G, Merchant Mark A, Sliwkowski Mark X, de Sauvage Frederic J, Seshagiri S (2013) Oncogenic ERBB3 Mutations in Human Cancers. *Cancer Cell* **23:** 603-617

Jaqaman K, Kuwata H, Touret N, Collins R, Trimble WS, Danuser G, Grinstein S (2011) Cytoskeletal control of CD36 diffusion promotes its receptor and signaling function. *Cell* **146:** 593-606

Jura N, Endres NF, Engel K, Deindl S, Das R, Lamers MH, Wemmer DE, Zhang X, Kuriyan J (2009) Mechanism for activation of the EGF receptor catalytic domain by the juxtamembrane segment. *Cell* **137:** 1293-1307

Kani K, Warren CM, Kaddis CS, Loo JA, Landgraf R (2005) Oligomers of ERBB3 have two distinct interfaces that differ in their sensitivity to disruption by heregulin. *The Journal of biological chemistry* **280:** 8238-8247

Kawashima N, Nakayama K, Itoh K, Itoh T, Ishikawa M, Biju V (2010) Reversible dimerization of EGFR revealed by single-molecule fluorescence imaging using quantum dots. *Chemistry* **16:** 1186-1192

Keating E, Nohe A, Petersen NO (2008) Studies of distribution, location and dynamic properties of EGFR on the cell surface measured by image correlation spectroscopy. *European biophysics journal : EBJ* **37:** 469-481

Kholodenko BN, Demin OV, Moehren G, Hoek JB (1999) Quantification of short term signaling by the epidermal growth factor receptor. *J Biol Chem* **274:** 30169-30181

Kholodenko BN, Hancock JF, Kolch W (2010) Signalling ballet in space and time. *Nature reviews Molecular cell biology* **11:** 414-426

Kitaura J, Song J, Tsai M, Asai K, Maeda-Yamamoto M, Mocsai A, Kawakami Y, Liu FT, Lowell CA, Barisas BG, Galli SJ, Kawakami T (2003) Evidence that IgE molecules mediate a spectrum of effects on mast cell survival and activation via aggregation of the FcepsilonRI. *Proceedings of the National Academy of Sciences of the United States of America* **100:** 12911-12916

Kleiman LB, Maiwald T, Conzelmann H, Lauffenburger DA, Sorger PK (2011) Rapid phospho-turnover by receptor tyrosine kinases impacts downstream signaling and drug binding. *Molecular cell* **43:** 723-737

Kozer N, Barua D, Orchard S, Nice EC, Burgess AW, Hlavacek WS, Clayton AH (2013) Exploring higher-order EGFR oligomerisation and phosphorylation--a combined experimental and theoretical approach. *Mol Biosyst* **9:** 1849-1863

Kusumi A, Murakoshi H, Murase K, Fujiwara T (2005a) Single-molecule imaging of diffusion, recruitment, and activation of signaling molecules in living cells. 123-152

Kusumi A, Nakada C, Ritchie K, Murase K, Suzuki K, Murakoshi H, Kasai RS, Kondo J, Fujiwara T (2005b) Paradigm shift of the plasma membrane concept from the two dimensional continuum fluid to the partitioned fluid: high-speed single-molecule tracking of memebrane molecules. *Annu Rev Biophys Biomol Struct* **34:** 351-378

Kusumi A, Sako Y (1996) Cell surface organization by the membrane skeleton. *Current opinion in cell biology* **8:** 566-574

Kusumi A, Sako Y, 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:** 2021-2040

Lambert NA (2010) GPCR dimers fall apart. *Sci Signal* **3:** pe12

Landau M, Fleishman SJ, Ben-Tal N (2004) A putative mechanism for downregulation of the catalytic activity of the EGF receptor via direct contact between its kinase and C-terminal domains. *Structure* **12:** 2265-2275

Lavi Y, Edidin MA, Gheber LA (2007) Dynamic Patches of Membrane Proteins. *Biophys J* **93:** L35-L37

Lavi Y, Gov N, Edidin M, Gheber LA (2012) Lifetime of major histocompatibility complex class-I membrane clusters is controlled by the actin cytoskeleton. *Biophys J* **102:** 1543-1550

Lee Y, Ma J, Lyu H, Huang J, Kim A, Liu B (2014) Role of erbB3 receptors in cancer therapeutic resistance. *Acta Biochimica et Biophysica Sinica* **46:** 190-198

Lemmon MA, Schlessinger J (2010) Cell signaling by receptor tyrosine kinases. *Cell* **141:** 1117-1134

Li H, Cao Y, Petzold LR, Gillespie DT (2008) Algorithms and software for stochastic simulation of biochemical reacting systems. *Biotechnol Prog* **24:** 56-61

Lidke DS, Wilson BS (2009) Caught in the act: quantifying protein behaviour in living cells. *Trends in cell biology* **19:** 566-574

Lillemeier BF, Mortelmaier MA, Forstner MB, Huppa JB, Groves JT, Davis MM (2010) TCR and Lat are expressed on separate protein islands on T cell membranes and concatenate during activation. *Nat Immunol* **11:** 90-96

Lillemeier BF, Pfeiffer JR, Surviladze Z, Wilson BS, Davis MM (2006) Plasma membrane-associated proteins are clustered into islands attached to the cytoskeleton. *Proceedings of the National Academy of Sciences of the United States of America* **103:** 18992-18997

Linderman JJ (2009) Modeling of G-protein-coupled receptor signaling pathways. *The Journal of biological chemistry* **284:** 5427-5431

Linggi B, Carpenter G (2006) ErbB receptors: new insights on mechanisms and biology. *Trends in cell biology* **16:** 649-656

Liu B, Ordonez-Ercan D, Fan Z, Edgerton SM, Yang X, Thor AD (2007) Downregulation of erbB3 abrogates erbB2-mediated tamoxifen resistance in breast cancer cells. *International Journal of Cancer* **120:** 1874-1882

Liu P, Cleveland TE, Bouyain s, Byrne PO, Longo PA, Leahy DJ (2012) A Single Ligand is Sufficient to Activate EGFR Dimers. *Proceedings of the National Academy of Sciences* **109:** 10861-10866

Lo HW (2010) Nuclear mode of the EGFR signaling network: biology, prognostic value, and therapeutic implications. *Discovery medicine* **10:** 44-51

Low-Nam ST, Lidke KA, Cutler PJ, Roovers RC, van Bergen en Henegouwen PM, Wilson BS, Lidke DS (2011) ErbB1 dimerization is promoted by domain co-confinement and stabilized by ligand binding. *Nature structural & molecular biology* **18:** 1244-1249

Lu C, Mi LZ, Schurpf T, Walz T, Springer TA (2012) Mechanisms for kinase-mediated dimerization of the epidermal growth factor receptor. *The Journal of biological chemistry* **287:** 38244-38253

Macdonald-Obermann JL, Pike LJ (2009) The intracellular juxtamembrane domain of the epidermal growth factor (EGF) receptor is responsible for the allosteric regulation of EGF binding. *The Journal of biological chemistry* **284:** 13570-13576

Macdonald-Obermann JL, Piwnica-Worms D, Pike LJ (2012) Mechanics of EGF Receptor/ErbB2 Kinase Activation Revealed by Luciferase Fragment Complementation Imaging. *PNAS* **109:** 137-142

Martin-Fernandez M, Clarke DT, Tobin MJ, Jones SV, Jones GR (2002) Preformed oligomeric epidermal growth factor receptors undergo an ectodomain structure change during signaling. *Biophys J* **82:** 2415-2427

Mayawala K, Gelmi CA, Edwards JS (2004) MAPK cascade possesses decoupled controllability of signal amplification and duration. *Biophys J* **87:** L01-02

Mayawala K, Vlachos DG, Edwards JS (2005a) Computational modeling reveals molecular details of epidermal growth factor binding. *BMC Cell Biol* **6:** 41

Mayawala K, Vlachos DG, Edwards JS (2005b) Heterogeneities in EGF receptor density at the cell surface can lead to concave up scatchard plot of EGF binding. *FEBS letters* **579:** 3043-3047

Mayawala K, Vlachos DG, Edwards JS (2006) Spatial modeling of dimerization reaction dynamics in the plasma membrane: Monte Carlo vs. continuum differential equations. *Biophys Chem* **121:** 194-208

Mi LZ, Lu C, Li Z, Nishida N, Walz T, Springer TA (2011) Simultaneous visualization of the extracellular and cytoplasmic domains of the epidermal growth factor receptor. *Nature structural & molecular biology* **18:** 984-989

Miura Y, Hanada K, Jones TL (2001) G(s) signaling is intact after disruption of lipid rafts. *Biochemistry* **40:** 15418-15423

Miyagi H, Maruyama IN (2010) Analysis of Ligand-Receptor Interaction on the Surface of Living Cells by Fluorescence Correlation Spectroscopy. *The Open Spectroscopy Journal* **4:** 28-31

Monsey J, Shen W, Schlesinger P, Bose R (2010) Her4 and Her2/neu tyrosine kinase domains dimerize and activate in a reconstituted in vitro system. *The Journal of biological chemistry* **285:** 7035-7044

Nagy P (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:** 4251-4262

Nagy P, Claus J, Jovin TM, Arndt-Jovin DJ (2010) Distribution of resting and ligand-bound ErbB1 and ErbB2 receptor tyrosine kinases in living cells using number and brightness analysis. *PNAS***:** 1-6

Nagy P, Jenei A, Kirsch AK, Szollosi J, Damjanovich S (1999) Activation-dependent clusterting of the erbB2 receptor tyrosine kinase deteced by scanning near-field optical microscopy. *Journal of cell science* **112:** 1733-1741

Niehaus AMS, Edwards JS, Plechac P, Tribe R (2008) Microscopic Simulation of Membrane Molecule Diffusion on Corralled Membrane Surfaces. *Biophys J* **94:** 1551-1564

Ober RJ, Ram S, Ward ES (2004) Localization accuracy in single-molecule microscopy. *Biophys J* **86:** 1185

Ogiso H, Ishitani R, Nureki O, Fukai S, Yamanaka M, Kim JH, Saito K, Sakamoto A, Inoue M, Shirouzu M, Yokoyama S (2002) Crystal structure of the complex of human epidermal growth factor and receptor extracellular domains. *Cell* **110:** 775-787

Orr G, Hu D, Ozcelik S, Opresko LK, Wiley HS, Colson SD (2005) Cholesterol dictates the freedom of EGF receptors and HER2 in the plane of the membrane. *Biophys J* **89:** 1362-1373

Orton RJ, Sturm OE, Vyshemirsky V, Calder M, Gilbert DR, Kolch W (2005) Computational modelling of the receptor-tyrosine-kinase-activated MAPK pathway. *The Biochemical journal* **392:** 249-261

Parton RG, Hancock JF (2004) Lipid rafts and plasma membrane microorganization: insights rom Ras. *Trends Cell Biol* **14:** 141-147

Pezzarossa A, Fenz S, Schmidt T. (2011) Probing Structure and Dynamics of the Cell Membrane with Single Fluorescent Proteins. *Fluorescent Proteins II*. Springer, Berlin, Heidelberg, pp. 185-212.

Pierce SK, Liu W (2010) The tipping points in the initiation of B cell signalling: how small changes make big differences. *Nature reviews Immunology* **10:** 767-777

Pike LJ (2003) Lipid rafts: bringing order to chaos. *J Lipid Res* **44:** 655-667

Plowman SJ, Muncke C, Parton RG, Hancock JF (2005) H-ras, K-ras, and inner plasma membrane raft proteins operate in nanoclusters with differential dependence on the actin cytoskeleton. *Proceedings of the National Academy of Sciences of the United States of America* **102:** 15500-15505

Popov AV, Agmon N (2001) Three-dimensional simulations of reversible bimolecular reactions: The simple target problem. *The Journal of Chemical Physics* **115:** 8921-8932

Pryor MM, Low-Nam ST, Halasz AM, Lidke DS, Wilson BS, Edwards JS (2013) Dynamic transition states of ErbB1 phosphorylation predicted by spatial stochastic modeling. *Biophys J* **105:** 1533-1543

Radhakrishnan K (1991) Combustion kinetics and sensitivity analysis. In *Numerical approaches to combustion modeling*, Oran ES, Boris JP (eds), pp 83-128. Washington D.C.: AIAA

Radhakrishnan K, Edwards JS, Lidke DS, Jovin TM, Wilson BS, Oliver JM (2009) Sensitivity analysis predicts that the ERK-pMEK interaction regulates ERK nuclear translocation. *IET Syst Biol* **3:** 329-341

Radhakrishnan K, Halasz A, McCabe MM, Edwards JS, Wilson BS (2012) Mathematical simulation of membrane protein clustering for efficient signal transduction. *Annals of biomedical engineering* **40:** 2307-2318

Radhakrishnan K, Halasz A, Vlachos D, Edwards JS (2010) Quantitative understanding of cell signaling: the importance of membrane organization. *Curr Opin Biotechnol* **21:** 677-682

Radhakrishnan R (2010) ErbB receptor mediated oncogenic signaling: Molecular systems biology through multiscale modeling and high-performance computing. *Abstracts of Papers of the American Chemical Society* **240**

Rao M, Mayor S (2005) Use of Forster's resonance energy transfer microscopy to study lipid rafts. *Biochim Biophys Acta* **1746:** 221-233

Reddy AS, Chilukuri S, Raychaudhuri S (2010) The network of receptors characterize B cell receptor micro- and macroclustering in a Monte Carlo model. *J Phys Chem B* **114:** 487-494

Resat H, Petzold L, Pettigrew MF (2009) Kinetic modeling of biological systems. *Methods Mol Biol* **541:** 311-335

Reynolds AR, Tischer C, Verveer PJ, Rocks O, Bastiaens PI (2003) EGFR activation coupled to inhibition of tyrosine phosphatases causes lateral signal propagation. *Nat Cell Biol* **5:** 447-453

Rosenbaum DM, Rasmussen SG, Kobilka BK (2009) The structure and function of G-protein-coupled receptors. *Nature* **459:** 356-363

Saffarian S, Li Y, Elson EL, Pike LJ (2007) Oligomerization of the EGF receptor investigated by live cell fluorescence intensity distribution analysis. *Biophys J* **93:** 1021-1031

Saikh SR, Edidin MA (2006) Membranes are not just rafts. *Chem Phys Lipids* **144:** 1-3

Sako Y, Kusumi A (1994) Compartmentalized structure of the plasma membrane for receptor movements as revealed by a nanometer-level motion analysis. *The Journal of cell biology* **125:** 1251-1264

Sako Y, Kusumi A (1995) Barriers for lateral diffusion of transferrin receptor in the plasma membrane as characterized by receptor dragging by laser tweezers: fence versus tether. *The Journal of cell biology* **129:** 1559-1574

Santamaria F, Gonzalez J, Augustine GJ, Raghavachari S (2010) Quantifying the effects of elastic collisions and non-covalent binding on glutamate receptor trafficking in the post-synaptic density. *PLoS Comput Biol* **6:** e1000780

Sasagawa S, Ozaki Y, Fujita K, Kuroda S (2005) Prediction and validation of the distinct dynamics of transient and sustained ERK activation. *Nat Cell Biol* **7:** 365-U331

Sato Y, Yashiro M, Takakura N (2013) Heregulin induces resistance to lapatinib-mediated growth inhibition of HER2-amplified cancer cells. *Cancer Science* **104:** 1618-1625

Saxton MJ (1993) Lateral diffusion in an archipelago. Single-particle diffusion. *Biophys J* **64:** 1766-1780

Saxton MJ (1995) Single-particle tracking: effects of corrals. *Biophys J* **69:** 389-398

Saxton MJ (2007) A Biological Interpretation of Transient Anomalous Subdiffusion. I. Qualitative Model. *Biophys J* **92:** 1178-1191

Saxton MJ (2008) A Biological Interpretation of Transient Anomalous Subdiffusion. II. Reaction Kinetics. *Biophys J* **94:** 760-771

Saxton MJ. (2009) Single Particle Tracking. *Fundamental Concepts in Biophysics*. Humana Press, Totowa, NJ, pp. 147-179.

Schlessinger J (2002) Ligand-induced, receptor-mediated dimerization and activation of EGF receptor. *Cell* **110:** 669-672

Schmidt T, Schütz GJ, Baumgartner W, Gruber HJ, Schindler H (1995) Photophysics and motion of single fluorescent molecules in phospholipid membranes. *J Phys Chem* **99:** 17662-17668

Schoeberl B, Eichler-Jonsson C, Gilles ED, Muller G (2002) Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nat Biotechnol* **20:** 370-375

Schütz GJ, Schindler H, Schmidt T (1997) Single-molecule microscopy on model membranes reveals anomalous diffusion. *Biophys J* **73:** 1073-1080

Sergina NV, Rausch M, Wang D, Blair J, Hann B, Shokat KM, Moasser MM (2007) Escape from HER-family tyrosine kinase inhibitor therapy by the kinase-inactive HER3. *Nature* **445:** 437-441

Shalom-Feuerstein R, Plowman SJ, Rotblat B, Ariotti N, Tian T, Hancock JF, Kloog Y (2008) K-ras nanoclustering is subverted by overexpression of the scaffold protein galectin-3. *Cancer Res* **68:** 6608-6616

Shankaran H, Wiley HS, Resat H (2006) Modeling the effects of HER/ErbB1-3 coexpression on receptor dimerization and biological response. *Biophys J* **90:** 3993-4009

Shankaran H, Zhang Y, Tan Y, Resat H (2013) Model-based analysis of HER activation in cells co-expressing EGFR, HER2 and HER3. *PLoS Comput Biol* **9:** e1003201

Shi F, Telesco SE, Liu Y, Radhakrishnan R, Lemmon MA (2010) ErbB3/HER3 intracellular domain is competent to bind ATP and catalyze autophosphorylation. *Proceedings of the National Academy of Sciences of the United States of America* **107:** 7692-7697

Simons K, Gerl MJ (2010) Revitalizing membrane rafts: new tools and insights. *Nature reviews Molecular cell biology* **11:** 688-699

Simons K, Ikonen E (1997) Functional rafts in cell membranes. *Nature* **387:** 569-572

Simson R, Sheets ED, Jacobson K (1995) Detection of temporary lateral confinement of membrane proteins using single-particle tracking analysis. *Biophys J* **69:** 989-993

Simson R, Yang B, Moore SE, Doherty P, Walsh FS, Jacobson KA (1998) Structural Mosaicism on the Submicron Scale in the Plasma Membrane. *Biophys J* **74:** 297-308

Singer SJ, Nicolson GL (1972) The Fluid Mosaic Model of the Structure of Cell Membranes. *Science* **175:** 720-731

Slepchenko BM, Schaff JC, Macara I, Loew LM (2003) Quantitative cell biology with the Virtual Cell. *Trends in cell biology* **13:** 570-576

Smith CS, Joseph N, Lidke KA (2010) Fast, single-molecule localization that achieves theoretically minimum uncertainty. *Nature methods* **7:** 373-375

Soderberg O, Gullberg M, Jarvius M, Ridderstrale K, Leuchowius KJ, Jarvius J, Wester K, Hydbring P, Bahram F, Larsson LG, Landegren U (2006) Direct observation of individual endogenous protein complexes in situ by proximity ligation. *Nature methods* **3:** 995-1000

Spira F, Mueller NS, Beck G, von Olshausen P, Beig J, Wedlich-Soldner R (2012) Patchwork organization of the yeast plasma membrane into numerous coexisting domains. *Nat Cell Biol* **14:** 640-648

Steinkamp MP, Low-Nam ST, Yang S, Lidke KA, Lidke DS, Wilson BS (2014) erbB3 Is an Active Tyrosine Kinase Capable of Homo- and Heterointeractions. *Molecular and cellular biology* **34:** 965-977

Suzuki K, Ritchie K, Kajikawa E, Fujiwara T, Kusumi A (2005) Rapid hop diffusion of a G-protein-coupled receptor in the plasma membrane as revealed by single-molecule techniques. *Biophys J* **88:** 3659-3680

Szabo A, Horvath G, Szollosi J, Nagy P (2008) Quantitative characterization of the large-scale association of ErbB1 and ErbB2 by flow cytometric homo-FRET measurements. *Biophys J* **95:** 2086-2096

Telesco SE, Radhakrishnan R (2012) Structural systems biology and multiscale signaling models. *Annals of biomedical engineering* **40:** 2295-2306

Telesco SE, Shih AJ, Jia F, Radhakrishnan R (2011) A multiscale modeling approach to investigate molecular mechanisms of pseudokinase activation and drug resistance in the HER3/ErbB3 receptor tyrosine kinase signaling network. *Mol Biosyst* **7:** 2066-2080

ten Klooster JP, Hordijk PL (2007) Targeting and localized signalling by small GTPases. *Biology of the cell / under the auspices of the European Cell Biology Organization* **99:** 1-12

Tian T, Harding A, Inder K, Plowman S, Parton RG, Hancock JF (2007) Plasma membrane nanoswitches generate high-fidelity Ras signal transduction. *Nat Cell Biol* **9:** 905-914

Tian T, Plowman SJ, Parton RG, Kloog Y, Hancock JF (2010) Mathematical modeling of K-Ras nanocluster formation on the plasma membrane. *Biophys J* **99:** 534-543

Tolle DP, Le Novere N (2010a) Brownian diffusion of AMPA receptors is sufficient to explain fast onset of LTP. *BMC systems biology* **4:** 25

Tolle DP, Le Novere N (2010b) Meredys, a multi-compartment reaction-diffusion simulator using multistate realistic molecular complexes. *BMC systems biology* **4:** 24

Turner TE, Schnell S, Burrage K (2004) Stochastic approaches for modelling in vivo reactions. *Comput Biol Chem* **28:** 165-178

Tynan CJ, Roberts SK, Rolfe DJ, Clarke DT, Loeffler HH, Kastner J, Winn MD, Parker PJ, Martin-Fernandez ML (2011) Human epidermal growth factor receptor (EGFR) aligned on the plasma membrane adopts key features of Drosophila EGFR asymmetry. *Molecular and cellular biology* **31:** 2241-2252

van den Bogaart G, Meyenberg K, Risselada HJ, Amin H, Willig KI, Hubrich BE, Dier M, Hell SW, Grubmuller H, Diederichsen U, Jahn R (2011) Membrane protein sequestering by ionic protein-lipid interactions. *Nature* **479:** 552-555

van Meer G, Simons K (1982) Viruses budding from either the apical or the basolateral plasma membrane domain of MDCK cells have unique phospholipid compositions. *EMBO J* **1:** 847-852

Varma R, Mayor S (1998) GPI-Anchored proteins are organized in submicron domains at the cell surface. *Nature* **394:** 798-801

Vaught DB, Stanford JC, Young C, Hicks DJ, Wheeler F, Rinehart C, Sánchez V, Koland J, Muller WJ, Arteaga CL, Cook RS (2012) HER3 Is Required for HER2-Induced Preneoplastic Changes to the Breast Epithelium and Tumor Formation. *Cancer Research* **72:** 2672-2682

Veatch SL, Machta BB, Shelby SA, Chiang EN, Holowka DA, Baird BA (2012) Correlation functions quantify super-resolution images and estimate apparent clustering due to over-counting. *PLoS One* **7:** e31457

Verveer PJ, Wouters FS, Reynolds AR, Bastiaens PI (2000) Quantitative Imaging of Lateral ErbB1 Receptor Signal Propagation in the Plasma Membrane. *Science* **290:** 1567-1570

Vigil D, Cherfils J, Rossman KL, Der CJ (2010) Ras superfamily GEFs and GAPs: validated and tractable targets for cancer therapy? *Nature reviews Cancer* **10:** 842-857

Vilardaga JP, Agnati LF, Fuxe K, Ciruela F (2010) G-protein-coupled receptor heteromer dynamics. *Journal of cell science* **123:** 4215-4220

Waller A, Sutton KL, Kinzer-Ursem TL, Absood A, Traynor JR, Linderman JJ, Omann GM (2004) Receptor binding kinetics and cellular responses of six N-formyl peptide agonists in human neutrophils. *Biochemistry* **43:** 8204-8216

Wang H, Rong G, Yan B, Yang L, Reinhard BM (2011) Optical sizing of immunolabel clusters through multispectral plasmon coupling microscopy. *Nano letters* **11:** 498-504

Warren CM, Landgraf R (2006) Signaling through ERBB receptors: Multiple layers of diversity and control. *Cell Signal* **18:** 923-933

Wells NP, Lessard GA, Goodwin PM, Phipps ME, Cutler PJ, Lidke DS, Wilson BS, Werner JH (2010) Time-resolved three-dimensional molecular tracking in live cells. *Nano letters* **10:** 4732-4737

Wennerberg K, Rossman KL, Der CJ (2005) The Ras superfamily at a glance. *Journal of cell science* **118:** 843-846

Wieser S, Moertelmaier M, Fuertenbauer E, Stockinger H, Sh\:utz G (2007) (Un)confined diffusion of CD59 in the plasma membrane determined by high-resolution single molecule microscopy. *Biophys J* **92:** 3719-3728

Wiley HS, Shvartsman SY, Lauffenburger DA (2003) Computational modeling of the EGF-receptor system: a paradigm for systems biology. *Trends in cell biology* **13:** 43-50

Wilson B, Pfeiffer JR, Oliver JM (2000) Observing FceRI Signalling from the Inside of the Mast Cell Membrane. *Journal of Cell Biology* **149:** 1131-1142

Wilson BS, Oliver JM, Lidke DS (2010) Spatio-temporal signaling in mast cells. *Adv Exp Med Biol* **716:** 91-106

Wilson BS, Pfeiffer JR, Raymond-Stintz MA, Lidke D, Andrews N, Zhang J, Yin W, Steinberg S, Oliver JM (2007) Exploring membrane domains using native membrane sheets and transmission electron microscopy. *Methods Mol Biol* **398:** 245-261

Wilson BS, Pfeiffer JR, Surviladze Z, Gaudet EA, Oliver JM (2001) High resolution mapping of mast cell membranes reveals primary and secondary domains of Fc\textepsilon RI and LAT. *J Cell Biol* **154:** 645-658

Yang S, Raymond-Stintz MA, Ying W, Zhang J, Lidke DS, Steinberg SL, Williams L, Oliver JM, Wilson BS (2007) Mapping ErbB receptors on breast cancer cell membranes during signal transduction. *Journal of cell science* **120:** 2763-2773

Yarden Y, Sliwkowski MX (2001) Untangling the ErbB signalling network. *Nat Rev Mol Cell Biol* **2:** 127-137

Yeow EK, Clayton AH (2007) Enumeration of oligomerization states of membrane proteins in living cells by homo-FRET spectroscopy and microscopy: theory and application. *Biophys J* **92:** 3098-3104

Zhang J, Steinberg SL, Wilson BS, Oliver JM, Williams LR (2008) Markov random field modeling of the spatial distribution of proteins on cell membranes. *Bulletin of mathematical biology* **70:** 297-321

Zhang K, Wong P, Duan J, Jacobs B, Borden EC, Bedogni B (2013) An ERBB3/ERBB2 oncogenic unit plays a key role in NRG1 signaling and melanoma cell growth and survival. *Pigment cell & melanoma research* **26:** 408-414

Zhang Q, Park E, Kani K, Landgraf R (2012) Functional isolation of activated and unilaterally phosphorylated heterodimers of ERBB2 and ERBB3 as scaffolds in ligand-dependent signaling. *Proceedings of the National Academy of Sciences of the United States of America* **109:** 13237-13242

Zhang XW, Gureasko J, Shen K, Cole PA, Kuriyan J (2006) An allosteric mechanism for activation of the kinase domain of epidermal growth factor receptor. *Cell* **125:** 1137-1149

Zhang Y, Opresko L, Shankaran H, Chrisler WB, Wiley HS, Resat H (2009) HER/ErbB receptor interactions and signaling patterns in human mammary epithelial cells. *BMC Cell Biol* **10:** 78

Zhdanov VP, Kasemo B (1994) Kinetic phase transitions in simple reactions on solid surfaces. *Surface Science Reports* **20:** 111-189