Johnson Presentation Script

Slide 1: The first thing I did here was implement my own version of a particle-based reaction diffusion simulation in python. Here I have 3 species, with A and B associating to form C. The model is determined by the parameters in the blue box. As you can see in the 3D plot, C in green occasionally gets created and then diffuses locally until it disassociates.

Slide 2: Here is the basic logic or pseudo-code of the simulation, which I broadly modeled on NERDSS own sequence of behaviors. Every time step or iteration, all complexed species are checked against a probability distribution to decide whether they remain for this iteration, or separate into two randomly generated vectors in opposite directions but equal magnitude representing liberated A and B. Next, all pairwise distances between A and B are evaluated and used to determine the pair’s unique association probability, which is a function of their distance. After dissociation and association checks are completed, all particles are moved in a Brownian fashion, with their new coordinate vector being chosen from a Gaussian bounded by a function of the diffusion constant and time step size. Finally, the boundary conditions for both the water box and inter-species collision is enforced, and coordinates are corrected, before the next iteration.

Slide 3: Here are the probability functions used to determine association and dissociation events. The association probability is just the complement of the particle survival probability, or the volume integral of the probability function for irreversible association, and gives the probability of a pair having associated in time dt given a starting distance r. The dissociation probability function is just defined as a Poisson process.

Slide 4: Here is a graph of the calculated equilibrium copy numbers for each species superimposed onto the graph from the simulation, and much to my surprise, there was a lot of agreement aside from the volatility around the equilibrium points, which is probably a result of a large diffusion constant. The math I used to calculate the theoretical equilibrium copy numbers is here, where I first convert to copy numbers, then use the mass conservation constraints here to substitute and isolate a single variable, where I chose A. Then just solve the quadratic and back substitute to produce all equilibrium copy numbers.

Slide 5: I thought this would be an interesting comparison to show. In my first couple versions of my simulation, I didn’t evaluate collisions, and as a result was getting disagreement between theory and simulation. After I implemented a collision detector, the excessive binding was mitigated as A and B weren’t able to occupy the same space and had to remain at least 2 sigma apart.

Slide 6: After implementing my own code in Python, I moved on to using NERDSS to model a system of reactions described in this paper, with the adaptor protein AP2 binding to the lipid membrane and then interacting with transmembrane proteins.

Slide 7: As shown in this cartoon, AP2 has multiple binding domains that allow it to bind to both clathrin as well as the lipid membrane and cargo receptors. Through this role it is crucial for most CME, as clathrin cannot bind to the membrane directly.

Slide 8: Here is my simple toy model of the process. AP2 in solution is bound and localized to the explicit lipid PIP2. The adaptor/lipid complex on the membrane than can perform a 2D bimolecular association with the membrane bound Transferrin receptor, which is the cargo receptor for Transferrin, which shuttles iron through the blood. Transferrin is not particular to this model, as it simply contains a signal peptide TGN 38 which AP2 binds with high affinity and was tested in the experimental paper.

Slide 9: I performed the same theoretical validation of my simulation as I did with my Python model, with good agreement.

Slide 10: The theoretical validation was again simply converting the equilibrium equations into copy numbers, which simplifies dealing in 3D or 2D, and substituting in the mass conservation constraints to isolate a. This expression was quartic, as it described two 2nd order reactions; I’d have shown it but it’s extremely long.

Slide 11: So back to the paper that was inspiration for this model; they were only able to report the Kdobs for the system of reactions, as they were able to measure the amount of AP2 that localized to the membrane. As Kd2 is effectively hidden from us, it would be interesting to find a relationship between the reported Kdobs and Kd2 for simulation input parameters. To first describe Kdobs, the assumption that PIP2 was the dominant regime was made, as PIP2 is what controls localization to the membrane in the first place and will effectively be in excess. This produces this equation, which after substitutions with Kdap and Kdt, produce a relationship between Kdobs and Kdt. This makes sense too, as you can see as [T] approaches zero, Kd2 cancels out and Kdobs is dominated by the first disassociation constant. To validate the arithmetic behind this equation, I generated simulations with differing amounts of [T]eq and calculated the simulated Kdobs with the original equation.

Slide 12: Here is the theoretical relationship between Kdobs and Kdt plotted as a function of T, and in the red are the simulated data points.

Slide 13: In the same vein, I found a relationship between how cargo binding, or the AP complex binding to T, increases residence times, which would also describe different Kdobs as a function of T. I made a few assumptions to produce this expression, with the first being that residence time is the reciprocal of the off rate for Kdobs, which will allow me to make the final expression a function of both reactions. I also say the forward rate constant for Kdobs is equal to that of the Kdap, as the first reaction is what controls localization to the membrane. I sub this equations into the previously derived equation for Kdobs, and produce the final expression for residence time.

Slide 14: I then test this against my simulations, but simulating systems with only one AP2 molecule and varying amounts of T, in order to track the same particle’s binding history over time. The averages are shown here in the colored data points superimposed on the function for residence time.

Slide 15: My next simulation added an additional species to the mix, the CD4 receptor, whose dileucine motif is also recognized by ap2 and tested in the paper.