Alife – Project Report

Candidate # 81222 (Ken Webb)
January 10, 2004

Abstract

Biologists are unsure of many details of how proteins in a cell are sorted in the Golgi complex before being transported to various other destinations. This paper describes a minimal individual-based simulation model that shows one way to achieve biological sorting, by using lipid domains of two different thicknesses. Membrane lipids are modeled as cellular automata (CA), membrane proteins as multiagents interacting with the lipids, and vesicle budding and docking as emergent phenomena. Experimental runs of the simulation, with manipulated independent variables and a measured dependent variable, are described.

Introduction

Eukaryotic cells (those found in all higher organisms) contain multiple compartments that each perform distinct functions (Alberts, 2002; Becker, 1996). The endoplasmic reticulum (ER) is a major site of protein production. These proteins are transported in small spherical membrane-bounded vesicles to the Golgi complex for sorting, a process that determines the proteins' final destinations. Some of the fine details of sorting are not yet known (Armstrong, 2003), such as the exact mechanism by which proteins become selectively attached to specific domains in the Golgi membrane bilayer. These regions subsequently bud off to become vesicles destined for the cell's outer plasma membrane.

The goal of this project is to produce a minimal model that demonstrates the emergence of protein sorting in the Golgi, given a set of simpler lipid and protein movement behaviors. This paper describes an implementation of a minimal model that I have produced using NetLogo (NetLogo itself, 1999).

Lipids as Cellular Automata

The lipids in the lipid protein model are represented by a cellular automaton (CA). Toffoli and Margolus (1986, p.221) identify two "fundamental virtues" of CAs that make them ideal for modeling a membrane consisting of a large number of lipids – CAs are inherently parallel and their actions are inherently local.

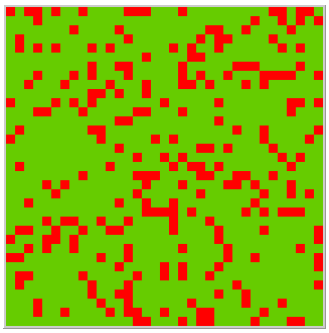

Figure 1 – Two types of lipids randomly positioned within a 2D 35 by 35 CA lattice. Each of the 1225 (35 x 35) cells is a NetLogo patch.

The lipid protein model uses a 2-dimensional (2D) discrete rectangular lattice, a 2D array of cells (Figure 1). A CA cell is called a patch in NetLogo and in the remainder of this paper. A patch represents a single lipid unit or molecule. Each patch has an x and y coordinate that represents its global location in the CA lattice.

The edges of the CA lattice wrap between bottom and top and between left and right edges. The lattice used in the lipid protein model is either 201 (x direction) by 201 (y direction) for a total of 201 * 201 = 40401 patches, or the NetLogo default of 35 by 35. The lipid protein model will run correctly with any square or rectangular lattice size.

Each patch in a CA has a local neighborhood (see Figure 2), adjacent patches that are positioned relative to the current patch. The lipid protein model uses a standard Moore local neighborhood (size = 8), implemented with the NetLogo neighbors primitive.

Figure 2 - A single patch (marked by X) surrounded by a local Moore neighborhood of 8 patches.

Time in a CA can be discrete or continuous, but is typically discrete measured in time steps. A typical NetLogo application has a function called go that is repeated forever. This is the approach followed in the lipid protein model.

All patches in a CA repeatedly follow the same rule or set of rules, at each time step. The rules depend on the current state of the other patches in the neighborhood. A patch in the lipid protein model can be either a Sopc type or a CholSopc type (Lundbæck et al., 2003). A Sopc patch contains exactly one unit or molecule of 1-stearoyl-2-oleoyl-phosphatidylcholine (SOPC). A CholSopc patch contains one unit of SOPC plus one unit of cholesterol (Chol). The thickness of a lipid unit and the color used to display it on the screen are both dependent on the lipidType, as follows:

lipidType	lipidThickness	lipidColor	
sopcType	3.0 nm	green	(light grey in the figures)
cholSopcType	3.3 nm	red	(dark grey)

The only possible operation for a lipid is to swap itself with an immediate neighbor of a different type. In a real cell, a fluid lateral diffusion occurs as lipids of all types constantly randomly swap places with other lipids in their local neighborhood, what is called the standard Fluid Mosaic Model (Becker, 1996, p.173).

Here is the algorithm used in the lipid protein model by each lipid patch at each time step:

```
If I am a CholSopc lipid Then
  If 0 of my 8 neighbors is a CholSopc lipid Then
    With probability ProbIf0, swap with a random one of my Sopc neighbors
  If 1 of my 8 neighbors is a CholSopc lipid Then
    With probability ProbIf1, swap with a random one of my Sopc neighbors
  …
  (same idea for 2, 3, 4, 5, and 6)
  …
  If 7 of my 8 neighbors is a CholSopc lipid Then
    With probability ProbIf7, swap with a random one of my Sopc neighbors
```

```
  If all 8 of my 8 neighbors are CholSopc lipids Then do nothing

If I am a Sopc lipid Then
  If fewer than 2 Of my 8 neighbors are Sopc lipids Then
    Swap with a random one of my CholSopc neighbors
```

Please refer also to the actual code in the *adjust-domain-size-continuous* function in the NetLogo program in the appendix.

This set of rules could alternatively be specified using a state-transition table. The lipid protein model uses a probabilistic or stochastic CA in which the next state is a function of a patch's current state, the conditions in the neighborhood, and a set of probability values. Each time step every lipid patch runs through this process and updates its internal state.

Budding, Docking, and Lipid Rafts in a CA

The lipid protein model also uses a larger extended neighborhood for a separate purpose. Lipid rafts (Simons & Ikonen, 1997) (see Figure 3) gradually emerge out of an initial random configuration and through the local actions of individual lipid patches, as described above. Lipid rafts are regions consisting largely of patches containing lipids of cholSopcType. They are high in cholesterol (Chol). These constructs emerge without being explicitly mentioned anywhere in the rules.

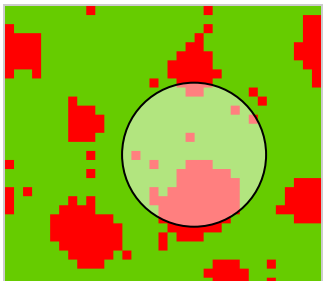

Figure 3 - Lipid rafts. The circular area has a BudRadius of about 7.

In the model, as a lipid raft grows in size, there is an increasing probability that a circular region of the membrane containing a raft will bud off to become a separate vesicle. The material that buds off is replaced by lipids in vesicles that dock with the membrane. A similar mechanism takes place in a real biological cell. Vesicles from the endoplasmic reticulum (ER) dock with the Golgi. Vesicles that are higher in cholesterol content later bud off from the Golgi and find their way to the plasma membrane on the outside of the cell.

An extended neighborhood is specified in NetLogo using the *in-radius* <*number*> primitive. The number of patches, including the current patch, that are within one radius unit, is 5. 13 patches are within a radius of 2, and 149 patches are within a radius of 7. The lipid protein model uses a default radius of 7 (BudRadius) to specify a region large enough to, with some probability, bud off.

Every n time steps (default: 1), a random patch is sampled to see if the number of lipids with lipidType = cholSopcType within this extended neighborhood exceeds some threshold (BudRadius). If so, a vesicle buds off. In the model this means that CholSopc lipids change their state to Sopc, and all proteins that were within this region cease to exist. The "lost" CholSopc lipids and proteins are recreated at random locations in the membrane, simulating

the docking of vesicles from the ER. The result is a reduction in order that counteracts the fine-grained order-increasing behavior of individual lipids described previously.

Proteins as Multiagents

Each of up to several thousand proteins in the lipid protein model is represented as a separate agent, with its own variable values and its own thread of control. NetLogo provides direct support for the multiagent paradigm. In biology, this type of model is often referred to as an individual-based model. Many well-known Alife models make use of a multiagent approach – bird flocking, ant systems, etc (Resnick, 1994).

Cellular automata, described in previous sections, provide a space and time background, "the computer scientist's counterpart to the physicist's concept of a field" (Toffoli & Margolus, 1987, p.5). A lattice of discrete patches represents space, while a sequence of discrete time steps represent time. In the lipid protein model, proteins move about within a lipid lattice field, and interact over time with the individual lipids in their immediate local neighborhood. An agent is called a turtle in NetLogo. Proteins are implemented as a specific breed (similar to the object-oriented concept of subclass) of turtle. The main practical value in modeling proteins as turtles rather than patches, is that (1) there are a relatively small number of proteins compared to the number of lipid patches, (2) proteins move, and (3) more than one protein may coexist at the same time within the same patch.

Proteins in the lipid protein model are simple abstractions. Their only attribute is a trans-membrane domain (TMD), a sequence of between 15 and 20 amino acids (AAs) (Lundbæck et al, 2003). This TMD is also called a hydrophobic ("water hating") alpha–helix, with a hydrophobic length L. A hydrophobic TMD seeks to embed itself within a region of lipid bilayer whose thickness is compatible with its hydrophobic length. A protein TMD with 20 amino acids has the largest hydrophobic length L, and seeks the thicker cholesterol-rich CholSopc lipid rafts. A protein TMD with only 15 amino acids has the shortest hydrophobic length L, and seeks the thinner cholesterol-free Sopc lipid regions.

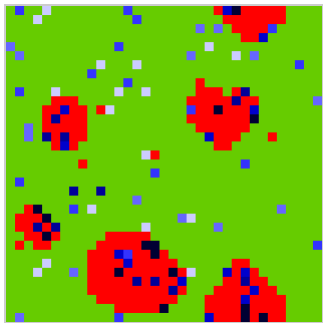

Figure 4 - Six types of protein multiagents embedded within a lipid bilayer CA. The longer (darker colored) protein TMDs are embedded within the thicker lipid rafts.

At each time step, all proteins follow the same set of rules in deciding whether to stay where they are or possibly move to a neighboring patch that may lead to a region of optimal lipid thickness. There are six protein types in the model – Tmd15, Tmd16, Tmd17, Tmd18, Tmd19, and Tmd20. The number refers to the number of amino acids in the TMD, and specifies the protein's hydrophobic length L (proteinL). The display color used on the

NetLogo lattice depends on the protein's proteinL value – Tmd20 is a very dark blue, Tmd15 is a light blue, while Tmd16 to Tmd19 are intermediate shades of blue. Here is the algorithm used by each protein at each time step:

```
If my TMD is partly exposed to water (i.e. my hydrophobic length > the
thickness of the lipid in which I am embedded) Then
  If a random one of my lipid neighbors is even thinner Then
    With very low probability P_ProbGtGt, move to that patch
  If a random one of my lipid neighbors is the same thickness Then
    With some probability P_ProbGtEq, move to that patch
  If a random one of my lipid neighbors is thicker Then
    With relatively high probability P_ProbGtLt, move to that patch

If my TMD is already optimal (i.e. my hydrophobic length = the thickness of
the lipid in which I am embedded) Then
  If a random one of my lipid neighbors is thinner Then
    With very low probability P_ProbEqGt, move to that patch
  If a random one of my lipid neighbors is the same thickness Then
    With some probability P_ProbEqEq, move to that patch
  If a random one of my lipid neighbors is thicker Then
    With relatively low probability P_ProbEqLt, move to that patch

If my TMD has room to spare ((i.e. my hydrophobic length < the thickness of
the lipid in which I am embedded) Then
  If a random one of my lipid neighbors is thinner Then
    With relatively high probability P_ProbLtGt, move to that patch
  If a random one of my lipid neighbors is the same thickness Then
    With some probability P_ProbLtEq, move to that patch
  If a random one of my lipid neighbors is even thicker Then
    With relatively low probability P_ProbLtLt, move to that patch
```

Please refer to the code in the *seek-optimal-site* function in the NetLogo program in appendix.

Figure 5 (see appendix for color) shows lipid patches in a CA, protein multiagents, and the higher level processes of budding and docking working together in the lipid protein model.

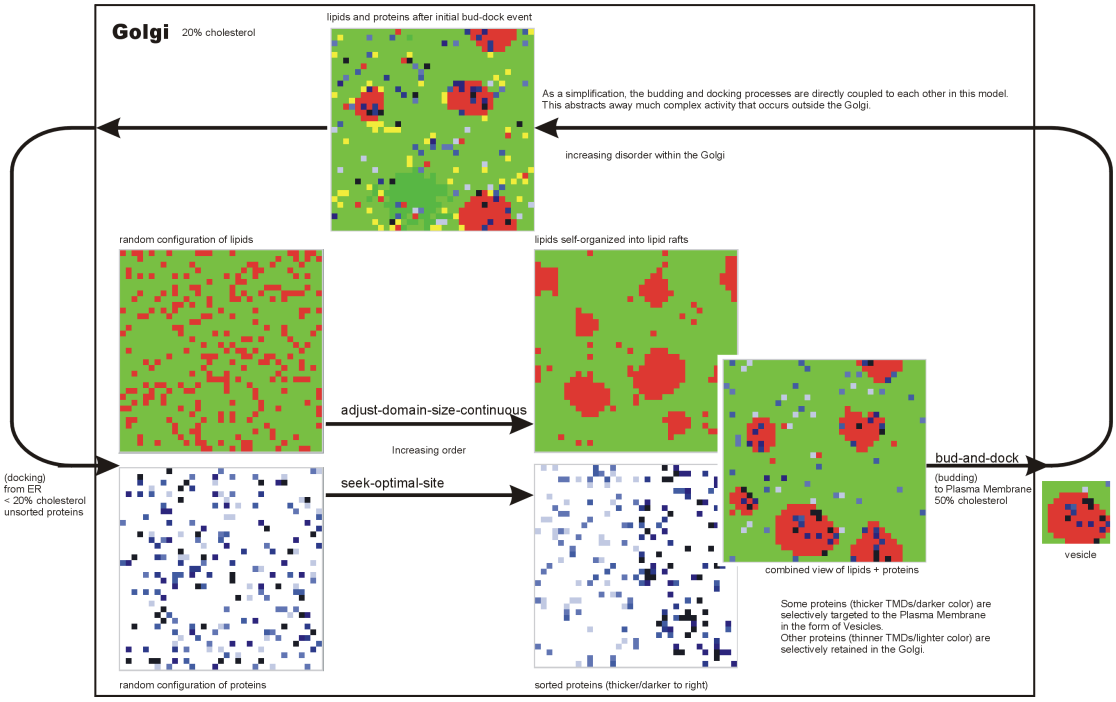

Figure 5 - Three processes at work that in combination produce continuous protein sorting.

Experiments with the Model and Critical Discussion

Initial experiments were done using two types of lipids, and six types of proteins corresponding to the six discrete TMD lengths found in real membranes. The model was then simplified to include only two discrete TMD lengths (short and long). This minimal model displays the same type of behavior as the more complex model. The experiments discussed in this report are those performed using this minimal model.

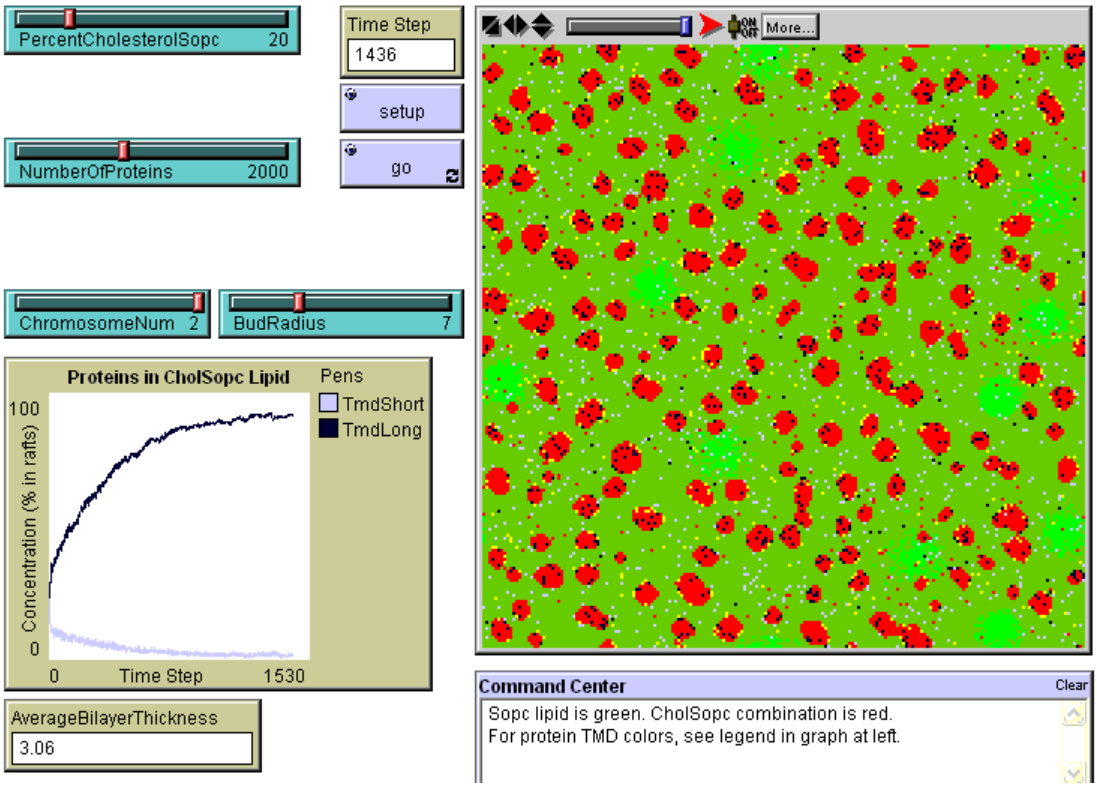

Figure 6 - Lipid Protein Model GUI showing the three independent variables ChromosomeNum, BudRadius, and (implicitly) GridSize. The CA lattice (grid) is 201 by 201. The measured dependent variables (percentage of TmdShort and TmdLong embedded within CholSopc lipids) is shown graphed over time.

There are three experimentally manipulated independent variables in the experimental runs – ChromosomeNum, BudRadius, and GridSize (see Figure 6 and Table 1). ChromosomeNum is a chromosome number, that indicates which of two lists of 18 movement probability values to use during that run. These lists are thought of as chromosomes because they could be manipulated using a genetic algorithm (GA) in a future version. For now, the values are hand coded and represent just two of a very large number of sets of possible integer values. Both chromosomes contain plausible values, but chromosome 2 imposes stricter rules on movement to less optimal neighbors. It encourages exploitation rather than exploration, while chromosome 1 is more focused on exploration. BudRadius can vary from 1 to 20, but only 5 through 8 produce a range of interesting measurable differences, with lower values more likely to produce budding. GridSize is the size of the 2D grid on which lipid patches and

proteins move. GridSize is either 201 by 201, or 35 by 35 (the NetLogo default). Results indicate that BudRadius needs to be adjusted downward if GridSize is decreased, because of an interaction between the two, which is largely an artifact of the experimental situation.

Independent variable	Range of values
ChromosomeNum	1, 2
BudRadius	5, 6, 7, 8
GridSize	35by35, 201by201

Table 1 - Independent Variables in Experimental Runs

Additional parameters can be set from the GUI (see Figure 6). PercentCholesterolSopc has been left constant at 20%, assumed to be the correct real world value. NumberOfProteins can be varied between 0 and 5000, but has little effect other than in the smoothness of the resulting curve (same mean, greater variance for low values).

The measured dependent variable in the experiments is the percentage of each type of protein (short-TMD and long-TMD) embedded within CholSopc lipids. This measure does not discriminate between solitary CholSopc patches and CholSopc found in lipid raft regions. As an example, in the graph in Figure 6, after 1436 time steps, over 90% of long-TMD proteins are embedded in CholSopc, while only about 4% of short-TMD proteins are embedded in CholSopc. This measure is used to demonstrate sorting of the two protein types.

The null hypothesis in the experiment is that an equal percentage of both short-TMD and long-TMD proteins will become embedded in lipid rafts. The experimental hypothesis is that there will be a significant difference in these percentages, thus demonstrating protein sorting.

Because of lack of time, proper experimental procedure has not been followed. There should for example be a certain number of sample runs for each experimental group in this 2 x 4 x 2 experiment (2 ChromosomeNum conditions by 4 BudRadius conditions by 2 GridSize conditions). No statistical analysis has been performed. The results reported here can therefore only be considered preliminary.

Informal results obtained by eye-balling the output graphs, suggest that there is a difference in the percentage of short-TMD proteins and long-TMD proteins that become embedded in CholSopc patches. There appears to be a difference across all experimental groups.

Initial experiments were conducted for the ChromosomeNum = 2, BudRadius = 7, and GridSize = 201by201 experimental group. As can be seen in Figure 6, there was initially a very large sorting effect. But the difference began to decline after about 4000 time steps (Figure 7 left), and the decline was continuing many hours later at time step 36500 (Figure 7

right) when the run was aborted. This combination of initial hyperbolic growth, followed by exponential decay was an initially surprising result. As time progresses, because of the

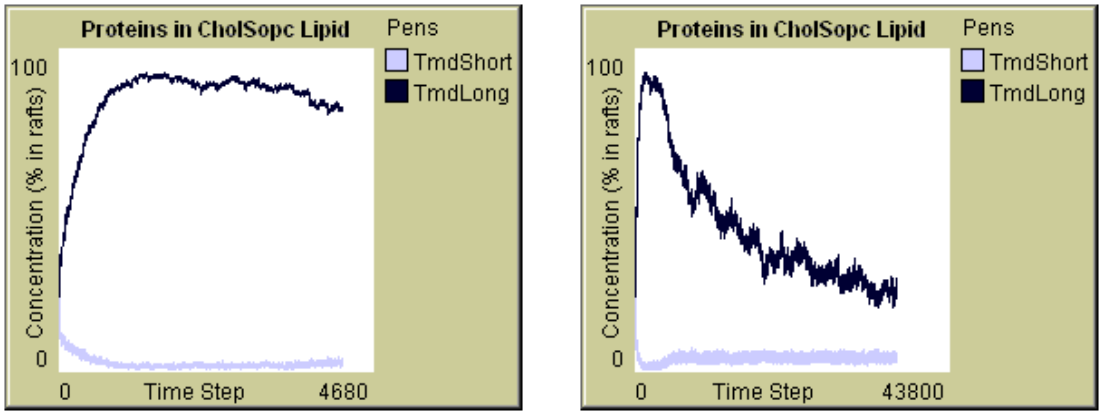

Figure 7 - Extent of % decline at TimeStep = 4229; and at TimeStep = 36500.

randomness introduced by many cycles of budding and docking, more and more of the long TMD proteins become transiently associated with individual CholSopc lipid patches that are not part of a cluster. These CholSopc are constantly moving so a high proportion of the long TMD are constantly being moved back into Sopc patches. Perhaps in the model each protein should with some probability move with the patch that it's currently located in.

Experiments for the ChromosomeNum = 1, BudRadius = 7, and GridSize = 201by201 group (Figure 8) produce a flatter curve lacking the boom and bust behavior, but these values do not produce clustering of CholSopc into lipid rafts sufficiently strong to trigger budding. Quite possibly there is a set of movement probability values that will produce sufficiently large lipid rafts at just the right rate to balance the opposing increases in order and disorder.

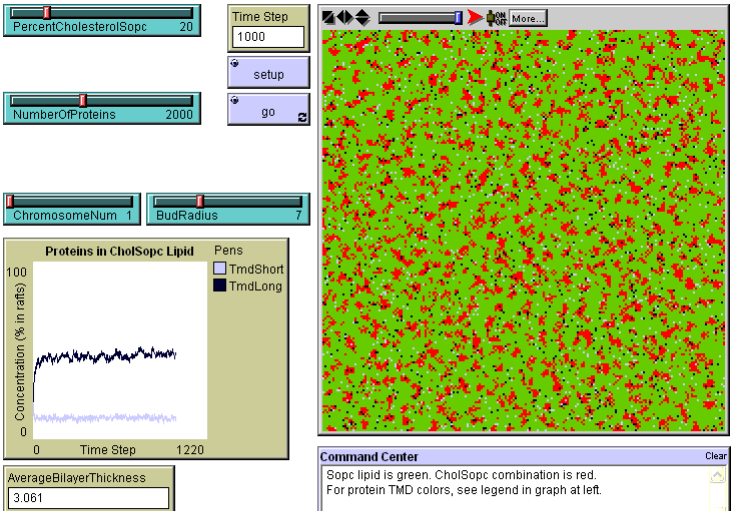

Figure 8 - Use of Chromosome 1 produces a flatter curve, but there is little clustering, no lipid rafts and no budding.

The NetLogo implementation is inefficient in both time and space, making it time-consuming to run simulations. For a 201 by 201 grid with 40401 patches and 2000 proteins, a single simulation run took about six hours to go through 36500 time steps, and consumed 100MB of memory. A smaller 35 by 35 grid that runs at least 10 times faster, produces results that appear qualitatively similar to those produced with the larger grid size (see Figure 9), although in this example budding produces wide swings because only 500 proteins are used.

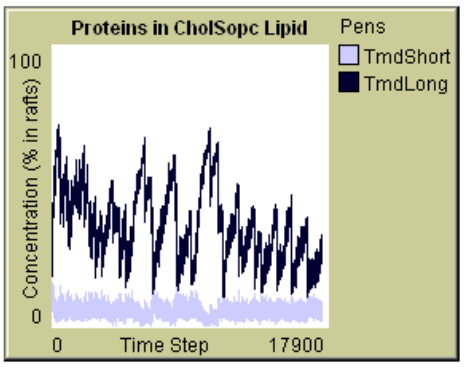

Figure 9 - ChromosomeNum = 2, BudRadius = 6, GridSize = 35by35, and num proteins = 500.

A number of runs with GridSize = 35by35, ChromosomeNum = 1, and BudRadius = 5, produce clustering, budding and a flat curve on the graph (see Figure 10) (see appendix for color version). However, the efficiency of protein sorting decreases using these parameters. Up to 11% of proteins in the budded vesicles are of the shorter type as opposed to only about 4% using the previous conditions (Chromosome 2 with a larger bud radius).

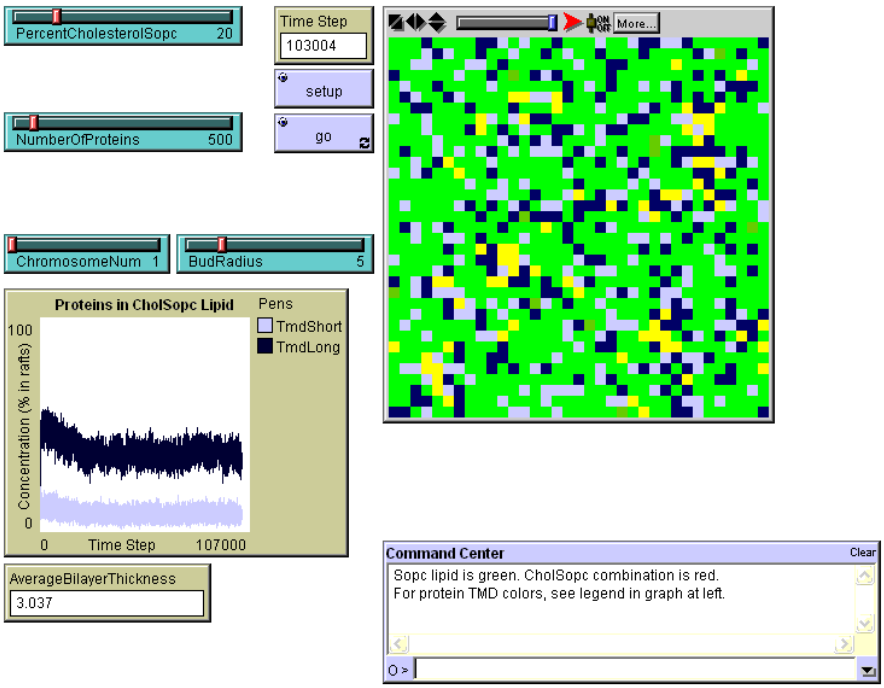

Figure 10 - A configuration that produces clustering into lipid rafts, budding, and a flat curve. (see appendix for color version).

This result would predict that a significant number of proteins with a shorter than expected TMD length should be observed at the plasma membrane of a cell, a result that could possibly be checked in a biology lab.

Possible Bugs

After a large number of runs, the average bilayer thickness decreases to around 3.04. The correct value should be (3.3*20 + 3.0*80) / 100 = 3.06 (CholSopcThickness * PercentCholesterolSopc + SopcThickness * (100 – PercentCholesterolSopc) / 100Percent). This may indicate a gradual loss of the thicker CholSopc, or a gain in the thinner Chol. This needs to be investigated.

There is also a synchronization issue, common to all CAs, unless proper care is taken. The model is not truely parallel, and consequently patches and proteins may be accessing neighboring values that are from different time steps. How this effects the model is not known. NetLogo always starts to update starting at the upper left corner of the grid, which could introduce some additional systematic bias.

Future Work

Two additional independent variables could be added in a future version:

1. There is some finite probability ($0.0 < p <= 1.0$) that the movement of a lipid patch will also move any given protein embedded within that patch.
2. Lipids and proteins arriving in vesicles from the ER are already somewhat clustered and pre-sorted.

Genetic algorithms (GAs) are often used to optimize some set of parameter values in a computer program. (Mitchell, 1998) The stochastic parameters in the lipid protein model have been arranged to make it relatively straight forward to derive using a GA. They have all been collected in one sequential list. This list can be thought of as a genome, each of whose genes can take on a range of possible values. When the values are copied into specific variables in the program, they can be thought of as specifying the behavior of a specific phenotype.

No attempt has yet been made to implement an actual GA to find optimal values in the lipid protein model. NetLogo would not easily support the concept of a population of membranes that would be needed to do this, nor does it have the processing speed required given that it is an interpreted language constructed using Java which is another interpreted language. To explore this parameter space using a GA, the lipid protein model should be rewritten in a faster and more flexible language such as C or C++.

The parameter values in the lipid protein model are integer rather than bit values. Each gene could be represented by a range of 16 integer values, which could be encoded using 4 bits. There are 19 genes in the system, so the entire genome could be represented in 4 * 20 bits, which is quite within the normal capabilities of a GA. Various encode/decode functions could be used depending on what actual values each parameter might take on. An arithmetic

encoding would produce the familiar 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 as possible phenotype values, while a geometric encoding would produce 0 1 4 9 16 25 36 49 64 81 100 121 144 169 196 225.

I note in passing that this involves two types of interacting entities within the same genome, and that this could be thought of as a type of co evolution between lipid behaviors and protein behaviors. This is a separate topic for future work.

Conclusion

The goal of producing a minimal model that demonstrates protein sorting in the Golgi has been met. The final model shown in Figure 10 exhibits:

1. The self-organization of lipids into domains, including thicker cholesterol-rich lipid rafts.
2. The selective association of protein trans-membrane domains (TMDs) with different lipid domains based on thickness, resulting in protein sorting.
3. The continuous budding off of vesicles from regions containing lipid rafts with associated sorted proteins as cargo, and replacement of this material through the docking of less self-organized vesicles.
4. Achievement of a steady-state in which the pathways of lipid self-organization, protein sorting, and vesicle budding and docking, collectively maintain a homeostatic organization, as shown in the flat curve of the graph in Figure 10.

This model also demonstrates some of the "guiding heuristics" described by Mitchell Resnick (1994, p.134), the original creative spirit behind what became NetLogo. Positive feedback and randomness combine to produce clustering into lipid rafts, in much the same way that these balance each other out in the slime mold model described by Resnick.

References Consulted

Adamatzky, A. (1994). Identification of Cellular Automata. London: Taylor & Francis.

Alberts, B., et al. (2002). Molecular Biology of the Cell, 4th ed. New York: Garland Science.

Armstrong, J. (2003). private communication.

Bagnat, M., et al. (2001). *Plasma Membrane Proton ATPase Pma1p Requires Raft Association for Surface Delivery in Yeast*. Molecular Biology of the Cell 12: 4129-4138.

Becker, W., et al. (1996). The World of the Cell, 3rd ed. Menlo Park, CA: Benjamin/Cummings.

Bretscher, M., and Munro, S. (1993). *Cholesterol and the Golgi Apparatus*. Science 261: 1280-1281.

Dumas, F., et al. (1999). *Is the protein/lipid hydrophobic matching principle relevant to membrane organization and functions?* FEBS Letters 458: 271-277.

Glick, B. (2000). *Organization of the Golgi apparatus*. Current Opinion in Cell Biology 12: 450-456.

Glick, B., et al. (1997). *Hypothesis: A cisternal maturation mechanism can explain the asymmetry of the Golgi stack*. FEBS Letters 414: 177-181.

Grimmett, G., and Stirzaker, G. (1992). Probability and Random Processes, 2nd ed. Oxford: Clarendon Press.

Gutowitz, H. (ed.) (1991). Cellular Automata – Theory and Experiment. Cambridge, MA: MIT Press. Reprinted from Physica D 45 (issues 1-3).

Lundbæck, J., et al. (2003). *Cholesterol-induced protein sorting: An analysis of energetic feasibility*. Biophysical Journal 84: 2080-2089.

McIntosh, H. (1990). *Wolfram's Class IV Automata and a Good Life*. Physica D 45: 105-121.

Miller, G. (1995). *Artificial Life as Theoretical Biology: How to do real science with computer simulation*. University of Sussex.

Mitchell, M. (1998). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press.

Munro, S. (1998). *Localization of proteins to the Golgi apparatus*. Trends in Cell Biology 8: 11-15.

NetLogo itself: Wilensky, U. 1999. NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

Poundstone, W. (1985). The Recursive Universe. New York: Morrow.

Raiborg, C., et al. (2003). *Protein sorting into multivesicular endosomes*. Current Opinion in Cell Biology 15: 446-455.

Resnick, M. (1994). Turtles, Termites, and Traffic Jams. Cambridge, MA: MIT Press.

Simons, K., and Ikonen, E. (1997). *Functional rafts in cell membranes*. Nature 387: 569-572.

Toffoli, T., and Margolus, N. (1987). Cellular Automata Machines. Cambridge, MA: MIT Press.

Van Meer, G. (1998). Lipids of the Golgi membrane. Trends in Cell Biology 8: 29-33.