Summary of ECM System Interactions

**Implementing an ECM**

This semester, our group was tasked with creating code for the existing PhysiCell framework that would allow for the simulation of an extracellular matrix. The extracellular matrix (ECM for short) is the fluid, waste, microfibers, and other materials that make up a cell’s microenvironment. In our simulation, we represented three qualities of the ECM: density, anisotropy, and fiber alignment. The ECM is created by instantiating one ECM class for each voxel of our microenvironment, each of which contains density, anisotropy, and fiber alignment values for that section.. The fiber alignment of each ECM index (?better word for this?) is stored in a singular double vector and normalized. These ECM data structures are stored a vector of ECM data objects.

An ECM class is instantiated for each voxel during the call to ECM\_setup in the main function. This setup function also randomizes the alignment of each ECM fiber and calls sync\_to\_BioFVM. To update the ECM that we created, we call the cell\_update\_from\_ecm during each iteration of our main loop, which in turn calls the cell altering functions which we describe later. This sets the time between ECM updates to the diffusion timescale of BioFVM.

**How Cells Change the ECM**

The extracellular matrix and the cells have a reciprocal relationship. In each iteration of the main loop we have a function called ecm\_update\_from\_cell. This larger update function calls three smaller functions that modify density, fiber alignment, and anisotropy. The first function, Cell-ECM density interaction, causes the cells to degrade the ECM density every time cell’s phenotypes are updated. This was created because as more cells move throughout the ECM they make it more porous. The second function, Cell-ECM fiber realignment, causes the fibers to slightly realign each time the cell’s phenotypes are updated. As cells continue to move throughout the ECM, they have a bias to move in the direction of the fibers. And as more cells move along the same gradient they realign the fibers in that specific direction, thus creating a positive feedback loop. The final function, Cell-ECM anisotropy modification, causes cells to increase ECM isotropy at rate proportional to their speed.

**How the ECM Changes Cells**

In each iteration of our loop in main, the cell\_update\_from\_ecm function is called. This in turn calls two more functions: change\_speed\_ecm and change\_bias\_ecm. The first function modifies the motility.migration\_speed property of the passed cell’s phenotype (if the cell is already motile) according to a parabolic function. We chose a parabolic function because too low of ECM density should result in cells lacking structure to grab onto that would allow them to move, and too high of a density should result in the cells being unable to overcome the force of the ECM holding them in place, removing their ability to be motile.

The change\_bias\_ecm function alters the migration\_bias and migration\_bias\_direction properties of a cell’s phenotype. The migration\_bias is simply set to be equal to the anisotropy of the ECM index corresponding to the inputted cell’s voxel. The migration\_bias\_direction vector is updated by component according to the equation (should I put the equation in or just explain it?). Before this is done, the dot product of *d* (the cell bias direction vector) and *f* (the fiber alignment vector) is calculated. If it is less than zero, then the sign of each component of *f* is flipped before it is used in the above equation. Finally, the cell’s updated bias direction is normalized before the function returns.

Sweet!!!