% Cell Lipid Model Version 2

% 3/29/18

% Niko Icardo

% Updates Version 2.2

% 1) Fixed Various errors and bugs

% - Instance of probArray was labeled propArray

%- Counter i in the probArray filling method was simultaneously used for keeping track of the position in the array and which cell’s rates were being added. If i was incremented by one, the cell would change to the next cell unintentionally adding its rates to the array rather than the appropriate cell’s rates. \* Changed the cell marker to be headcount instead of i, adjusting the index calls and the while loop condition to reference that marker. Reset headCount above the loop. Incremented headcount at the bottom of the loop to account for shifting the perspective to the next cell. Note, the loop still does not account for cell death\*.

%- Matlab returns decimal values for integer division. Thus, in the event selection section, the cellSelect assign, a decimal was given rather than a supposed integer, which would return an error when that cell was called from the cellArray consistent only of integer values. Changed the cellSelect math to round up the decimal to the nearest integer.

%- Two instances of cellSelect were labeled selectCell. Relabeled selectCell instances to the appropriate cellSelect variable.

%- Events were improperly labeled 1,2 and 0, with 1 corresponding to 0, 1 to 2 and 3 to 2. Changed the even selection value to the appropriate values consistent with the modulus of i in the event selection: 1 = 1, 2 = 2 , 3 = 0.

%- Changed a comment error in the new cell generation code in the event handling section. Lipids was incorrectly labeled mass.

%- Corrected various comment errors, and incorrect information.

%------------------------------------------------------------------ 4/5/18

%- In the Markov Dynamics Section, the initial cell instantiation was done incorrectly. Fixed.

%- Initialization of ProbArray was done incorrectly. According to Matlab documentation, zeroes(300) will create a 300x300 array not a 300x1 array as assumed. Fixed, as well as all instances of the probArray’s usage.

%- The previous version didn’t update every future value of the cell when one of its values were updated. This was remedied by extending the cell event value updater to time steps j:1000 from j.

%------------------------------------------------------------------- 4/12/18

%- The random value generation for u2 ( an event selection parameter) was erroneous. This was repaired: Find in the “calculate random values section”

%- The select event handling variables while loop was unnecessarily large. I changed it by reordering the steps and removing the if statement.

%- The lipid dependent mass generation equation had the initial value for lipids as the initialized variable while later it was changed to the cell I’s value for lipids. I changed this so that it will immediately search for cell i’s value when it is used in the rateTotal. Consequentially, the calculation in the rateTotal section could be removed and was.

%- u2 Was being generated between zero and rateTotal not zero and 1 like it should have been.

% Remarks

% 2) If cell death is implemented, the probArray method of allocating

% probabilities will have to be adjusted. It currently ends when a cell location isnt filled while this may lead to cells listed after that location not being accounted for if cell death is implemented.

% Variables

% Markov dynamics variables

u1 = 0; % The exponential random variable used in the time increment calculation

u2 = 0; % The random variable used to determine event occurrence

mu = 1; % The mean parameter used in generating the exp. random variable u1. It is set to one over the rateTotal per time step;

j = 1; % Time step counter for Markov model

rateTotal = 0; % this will become the sum of all rates for use in the selection of the time increments

i = 1; % a loop counter

headCount = 0; % A loop variable to ensure that every living cell is counted during the rate calculation and probability array generation

% Probability Array

probArray = zeros(300,1); % Stores the probability for each cell process to aid in event selection.

tempProbSum = 0; % Keeps a record of the sum of all probabilities calculated as of increment i in the probability array loop to aid in assigning values

% Event Handling Variables

cellSelect = 1; % This is used to mark which cell has an event occur.

event = 1; % This marks what type of event occurs for the aforementioned cell.

% Environmental variables

eCarbon = 100; % external carbon

eNitrogen = 100; % external nitrogen

initCellCount = 10; % Starting cell count

cellLimit = 100; % the maximum amount of cells allowed in the environment, intended to be a termination point of the simulation

cells = zeros(1000,cellLimit,3); % This array documents each time step j ( set at an arbitrary maximum ), how many cells are alive for which the maximum is j, and their two respective characteristics mass and lipid count,(the first page is for cells, second for mass, third for lipids).

population = initCellCount; % This is the current cell count. Initialized to the initCellCount.

t = (0:.1:1000); % Delta t

lastCellLocation = initCellCount; % this is used to assist in the addition of new cells to the cell matrix

%Cell variables

% Cell characteristics

mass = 5; % Cell's mass: the initial value for each cell is given

cellDivide = 10; % The mass limit of the cell that induces cell division

lipids = 2; % Cell's lipids: the initial value for each cell is given

Ymc = 1; % Yield of mass production on carbon

Yml = 1; % Yields of mass production on lipids

Ymn = 1; % Yields of mass production on nitrogen

Ylc = 1; % Yields of lipid production on carbon

% Nutrient dependent rate of mass production

nMassProdMax = 1; % Max rate of mass production on nutrients

Kc = 1; % Nutrient mass production carbon limiting constant

Kn = 1; % Nutrient mass production nitrogen limiting constant

nMassProd = (nMassProdMax)\*(eCarbon/(eCarbon +Kc))\*(eNitrogen/(eNitrogen +Kn));

% Lipid-dependent rate of mass production

lMassProdMax = 1; % Max rate of mass production on lipids

Kml = 1; % Lipid mass production lipid limiting constant

k1 = 1; % Lipid mass production rate adjustment parameter

lMassProd = (lMassProdMax)\*((cells(j,i,3)/(cells(j,i,3)+Kml))\*(k1/(eNitrogen+k1)));

% This is the individual dependent equation for lipid based mass generation.

% Nutrient-dependent rate of lipid production

LipidProdMax = 1; % max rate of lipid production

k2 = 1; % What is this?

Klc =1; % Lipid production limiting constant

lipidProduction = (LipidProdMax)\*(eCarbon/(eCarbon+Klc))\*(k2/(eNitrogen + k2));

% Carbon Consumption

totalCarbonConsumption = (nMassProd/Ymc) + (lMassProd/Ylc); % This will be looped for all cells to update the eCarbon Level

% Lipid Consumption

lipidConsumption = lMassProd/Yml;

% Markov Dynamics

% Initial cell instantiation

cells(1:1000,1:initCellCount,1) = 1; % 1 indicates a living cell, 0 a dead cell. Generate all living cells.

cells(1:1000,1:initCellCount,2) = mass; % Initial Cells' initial mass value is set

cells(1:1000,1:initCellCount,3) = lipids; % Cell initial mass and lipid values are set

%Main Loop

while(population < cellLimit && population > 0 && j < 1000) % Begin Markov Model simulation

% Calculate the sum of the rates and input them into the rate probability

% array

rateTotal = 0; % Reset rate total so that it may be calculated from new

i = 1; % Reset counter for rateTotal loop , index starts at 1

headCount = 0; % Reset headcount for rateTotal loop

probArray = zeros(300,1); % Reset all rates assigned previously

while( headCount < population) % A loop to set the rate total

if cells(j,i,1) == 1 % Check to see if cell is alive

% Add each of cell i's rates to rateTotal

rateTotal = rateTotal + nMassProd + lMassProd + lipidProduction; % Add each of the cells rates to the total

headCount = headCount + 1; % Count the living cells

end

i = i + 1; % Increment i

end

i = 1; %Reset i for probArray Loop

tempProbSum = 0; % Reset tempProbSum for probArray Loop

headCount = 1; % Reset headcount to be used in the probArray loop (it has to start at one to satisfy matrix dimensions

while(headCount <= population + 1) % Calculate the rate probabilities for each cell ( three rates per cell )

tempProbSum = tempProbSum + nMassProd/rateTotal; % increase the prob by the ith cell's nMassProd event prob

probArray(i,1) = tempProbSum; % Set the nMassProd probability into the probArray

i = i + 1; % increment i so that the next rate can be placed in the next index location in the probArray

lMassProd = (lMassProdMax)\*((cells(j,headCount,3)/(cells(j,headCount,3)+Kml))\*(k1/(eNitrogen+k1)));

tempProbSum = tempProbSum + lMassProd/rateTotal; % increase the prob by the ith cell's lMassProd event prob

probArray(i,1) = tempProbSum; % Set the lMassProd probability into the probArray

i = i + 1; % increment i so that the next rate can be placed in the next index location in the probArray

tempProbSum = tempProbSum + lipidProduction/rateTotal; % increase the prob by the ith cell's lipidProduction event prob

probArray(i,1) = tempProbSum; % Set the lipidProduction probability into the probArray

i = i + 1; % increment i for the next cell’s set of rates

headCount = headCount + 1; % increment headcount so that next cell is now in focus

end

% Calculate the random values

mu = (1/rateTotal); % set the mean value of the exprand value to the current rateTotal

u1 = exprnd(mu); % Select an exponentially random variable to increment the time ??

u2 =((1000)\*rand(1,1))/1000; % Select a random number between 0 and the and 1000. Dividing that number by 1000 allows decimal values to account for decimal differences in rates

% Increment time

t(j+1) = t(j) + (-log(u1)/rateTotal); % Increment the time

% Select Event handling variables

i = 1; % reset i for event handling loop

cellSelect = 1; % Set the base value for the cellSelect variable

event = 1; % set the base value for the event variable

while( u2 > probArray(i,1)) % Start the loop to calculate the cell and event, if cell 1 event 1 is less than the the random value

i = i + 1;

end

cellSelect = ceil((i - 1)/3); % Gives the appropriate cell

event = mod(i,3); % Gives the appropriate event

% Handle event

if event == 1 % increase a cells mass by nutrient based generation

cells(j:1000,cellSelect,2) = cells(j,cellSelect,2) + 1; % increment the cell's mass by the mass yield.

eCarbon = eCarbon - 1/Ymc; % Decrease the environment carbon supply by the amount necessary to increase mass

eNitrogen = eNitrogen - 1/Ymn; % Decrease the environment nitrogen supply by the amount necessary to increase mass

if cells(j,cellSelect,2) >= cellDivide % if the cell is capable of producing another cell by division, then enact that

lastCellLocation = lastCellLocation + 1; % increment the last-cell location of the matrix

cells(j:1000,lastCellLocation,1) = 1; % add the cell the cell matrix

cells(j:1000,lastCellLocation,2) = cells(j,cellSelect,2)/2; % Give the new cell half of its parent's mass

cells(j:1000,lastCellLocation,3) = cells(j,cellSelect,3)/2; % Give the new cell half of its parent's lipids

cells(j:1000,cellSelect,2) = cells(j,cellSelect,2) - cells(j,cellSelect,2)/2; % subtract the new cells mass from the original

cells(j:1000,cellSelect,3) = cells(j,cellSelect,3) - cells(j,cellSelect,3)/2; % subtract the new cells lipids from the original

population = population + 1; % This updates the population count

end

end

if event == 2 % increase a cell's mass by lipid based generation

if cells(j,cellSelect,3) >= 1 %% This checks that there are lipids to use for mass generation, if there is, the process proceeds

cells(j:1000,cellSelect,2) = cells(j,cellSelect,2) + 1; % increment the cell's mass

cells(j:1000,cellSelect,3) = cells(j,cellSelect,3) - 1/Yml; % Decrease the cell's lipid supply by the amount necessary to increase mass

if cells(j,cellSelect,2) >= cellDivide % if the cell is capable of producing another cell by division, then enact that

lastCellLocation = lastCellLocation + 1; % increment the last-cell location of the matrix

cells(j:1000,lastCellLocation,1) = 1; % add the cell the cell matrix

cells(j:1000,lastCellLocation,2) = cells(j,cellSelect,2)/2; % Give the new cell half of its parent's mass

cells(j:1000,lastCellLocation,3) = cells(j,cellSelect,3)/2; % Give the new cell half of its parent's lipids

cells(j:1000,cellSelect,2) = cells(j,cellSelect,2) - cells(j,cellSelect,2)/2; % subtract the new cells mass from the original

cells(j:1000,cellSelect,3) = cells(j,cellSelect,3) - cells(j,cellSelect,3)/2; % subtract the new cells lipids from the original

population = population + 1; % This updates the population count

end

end

end

if event == 0 % Generate Lipids by use of carbon

cells(j:1000,cellSelect,3) = cells(j,cellSelect,3) + 1; % Increment the cell's lipids by the lipid yield

eCarbon = eCarbon - 1/Ylc; % Decrease the environment carbon supply by the amount necessary to increase lipids

end

j = j + 1; % increment the step

end

disp(cells(:,:,:));