Introduction

Cancerous tumours are dense populations of uncontrollably dividing cells. All cells require oxygen and nutrient supply as well as waste removal, functions provided by the circulatory system. Therefore, tumour growth is affected by vascularity. In order to obtain the blood supply necessary to develop into large, malignant tumours, cancer cells are able to induce blood vessel growth; cancerous tumours induce angiogenesis. The dependency of tumours on vasculature suggests angiogenesis as a target for medical intervention. However, tumour-induced angiogenesis is complex and not fully understood. It consists of multiple, interrelated feedback and feedforward mechanisms leading to unpredictable and counter-intuitive results [1]. In order to facilitate the search for appropriate mechanisms to target therapeutically, researchers are developing a comprehensive in silico tumour angiogenesis model [1]. A well-behaved model would also help predict treatment outcomes.

The tumour angiogenesis model actually consists of two separate models, one of angiogenesis and one of tumour growth. Advancements to the ultrasound imaging modality, specifically optimization of power-Doppler and contrast-ultrasound processing [1], have improved angiogenic growth monitoring (improvements to power-Doppler processing allow for better blood volume estimation, while those to contrast-ultrasound processing primarily increase robustness to motion). Improved angiogenic growth modeling has in turn helped in the development of the angiogenesis model.

The tumour growth model, the focus of this project, is based on the work of Enderling et al. [2]. A three-dimensional version of the model has been implemented in MATLAB (Appendix I). The model outlines four fundamental parameters affecting growth, while the inherent limitation of in silico implementation introduces an additional two parameters affecting growth. Before the tumour model can be integrated with the angiogenesis model these parameters must be tuned so that the tumour model reflects realistic growth patterns. Realistic growth patterns were established by using ultrasound to measure the growth of translated human breast cancer tumours in mice. This project uses batch processing to systematically search for appropriate parameter settings which match the growth behavior of the in silico to data obtained empirically from in vivo tumours. Quality of match is assessed using a cost function and verified using visualization.
Theory
The tumour model implemented in this project was proposed and outlined in a paper published by Enderling et al.*[2]. They simplify the complex behavior of tumour growth into a handful of simple relationships and a minimal number of parameters. The resulting iterative model is straightforwardly implemented in silico.

The Enderling model represents tissue space as a three-dimensional lattice (when implemented in code, a three-dimensional array). Each point in the lattice (each array entry) represents a cell. The model includes only cancerous tumour cells (healthy tissue cells surrounding the tumour are excluded). However, the tumour is heterogeneous, consisting of two different cancer cell types. Some cells are stem cells, which are immortal and can divide an infinite number of times. The other cells are regular cancer cells, which are not immortal and can divide only a finite number of times.

The model is iterative and at each time step each cell, be it stem type or regular type, interacts with adjacent lattice points. However, if all adjacent lattice points are occupied (Figure 1), the cell is quiescent and makes no actions. Cells can remain quiescent indefinitely, until adjacent lattice points become vacant.

Figure 1: Cancer cells are modeled as points in a lattice. Each iteration cells interact with vacant adjacent lattice points. If all adjacent lattice points are occupied the cell is quiescent. Adapted from H. Enderling et al., Br. J. Cancer 100:1917-1925, 2009.

Cells with vacant adjacent lattice points can migrate. The maximum distance cells can migrate (measured in lattice points, which is essentially equivalent to cell lengths) is specified by the parameter \( \mu \). Additionally, every iteration each cell divides into two daughter cells (again, provided space). While most new cells are of regular type, there is a small probability (1-10%) that a daughter cell will be of stem type. This probability is specified by the parameter \( p_s \). The significance of stem cells is that they continue to divide indefinitely. Regular cells, on the other hand, divide a pre-specified number of times and then die. This pre-specified number of divisions is the proliferation capacity, represented by parameter \( \rho \). Note that daughter cells are not reset to \( \rho \) but begin with the same number of divisions remaining as their parent decremented by one. Lastly, there is a small probability that any cell can spontaneously die, specified by \( \alpha \). Cells that die, either by expiration of proliferation capacity or spontaneously are simply removed (the lattice point made vacant).

The four model parameters are summarized in Table 1, while the model algorithm is summarized in Figure 2.
Table 1: Four parameters governing tumour growth outlined.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_s$</td>
<td>probability of stem cell</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>probability of spontaneous death</td>
</tr>
<tr>
<td>$\rho$</td>
<td>proliferation capacity</td>
</tr>
<tr>
<td>$\mu$</td>
<td>maximum migration distance</td>
</tr>
</tbody>
</table>

Enderling et al. implemented and tested their model and found that the parameters actually govern growth behavior quite counter intuitively. Instinctively one might presume that tumour size would be proportional to proliferation capacity, inversely proportional to spontaneous death, and negligibly affected by migration. However, Enderling et al. found that migration distance was actually the strongest determinant of tumour size while high proliferation capacity and low spontaneous cell death could actually hinder growth. Each stem cell rapidly divides into a mass of regular cells; essentially, each stem cell grows into a sub-tumour. However, these stem cell based tumours quickly reach steady state (Figure 3) and there is no threat of them becoming malignant. The balance of proliferation capacity (not too high) and spontaneous death (not too low) increases the probability that there will be space for a new stem cell to be generated. A long migration distance increases the probability that the new stem cell will “escape” to seed its own sub-tumour. Enderling et al. summarized the idea by deeming that tumours are conglomerate of self-metastases; tumours are essentially a collection of smaller tumours (Figure 4).
Implementing the model in silico introduces two new constraints. Iterative models can quickly become computationally expensive. The longer the simulation runs (i.e. the greater the number of iterations) the larger the tumour grows. Since the model must perform proliferation, migration, and apoptosis for each cell individually, the amount of time required for each iteration increases exponentially. Furthermore, each iteration the algorithm must search through the entire model lattice. Assuming a cube, increasing lattice dimensions by N increases the total number of lattice points by $N^3$, and rapidly increases computation time. Lastly, when performing migration the algorithm must search a region which is $(2\mu+1)^3$ for each individual cell, and again computation time increases rapidly with $\mu$. Due to the computational limitations, the scale of feasible in silico models is significantly smaller than that of real tumours, so comparison will require normalization.

Two discrete curves (model predicted growth versus empirically measured growth) can be compared using a cost function. Numerous cost functions exist, but one of the most common is sum of squared differences (Equation 1). The square of the differences between values is taken at each discrete step (i.e. time interval or iteration). The squared differences are summed, such that a low cost indicates good fit. Squaring values serves two functions. First, it ensures the absolute value of the difference. Secondly, it weights larger deviations more heavily than smaller deviations.
Methods

1. Tumour volume curves were obtained from data provided by Dr. Lacefield and Matthew Lowerison. The data was obtained by injecting mice with seed tumours of MDA-MB-231-lucD3H2LN, a luciferase expressing strain of the MDA-MB-231 breast cancer line. Over the course of eight weeks, tumour volume in eight mice was monitored using ultrasound.

2. Analysis was performed on the volume curves to obtain a target curve. The data was averaged at each time interval and a polynomial was fit to the set of average volumes.

3. The MATLAB script implementing the Enderling model (written by Mathew Lowerison) was modified into a function, so that the simulation could be run multiple times from within another script.

4. Based on the computational ideas outlined under “Theory”, it was decided to hold the number of iterations and lattice size constant at 100 and 100x100x100 respectively. Since empirical data was collected for 50 days, 100 iterations corresponds to a mitotic period of 0.5 days. Additionally, a 100x100x100 matrix does not represent the actual number of cells in real tumours. Therefore it was necessary to normalize all curves to one, providing a basis for comparison (we are primarily interested in curve shape). Note that for the \textit{in silico} model we make the assumption that volume is directly proportional to number of cells.

5. The idea of tumours being conglomerates of smaller tumours was explored. First the effect of parameter tuning on the volume of the stem cell seeded “sub-tumours” was investigated. Probability of stem cell was fixed at zero while migration distance was fixed at two. Proliferation capacity and spontaneous death were varied systematically. The model was run five times for each parameter condition and the resulting volumes were averaged. See Appendix II for script which batch processes these simulations.

6. Next the shape of the volume curve (growth pattern) was tuned. Based on the previous two steps, four parameters (number of iterations, lattice size, proliferation capacity, and spontaneous death) were fixed. Probability of stem cell and migration distance were varied systematically. A cost based on the sum of squared differences between the model volume curve and the empirical volume curve was calculated for each parameter condition. The model was run a total of five times for each parameter condition and the resulting costs were averaged. See Appendix III for script which batch processes these simulations.

7. Curves and tumours from previous step were visualized to verify results.
Results

Figure 5: Volume of tumour in eight mice was tracked over 50 days. Data points indicate average volume at each measurement interval (error bars indicate standard deviation). Curves were fitted to data: exponential (shown in red) and cubic polynomial (shown in green, with equation $y = 5E-06x^3 + 0.0002x^2 - 0.008x + 0.0469$).

Figure 6: Example of ultrasound image of mouse tumour used to determine volume.

Figure 7: Probability of stem cell is fixed at zero (we are looking at one “sub-tumour”). Migration distance is fixed at two. Proliferation capacity and probability of spontaneous death are varied to determine the contribution of each stem cell seeded “sub-tumour”. Each bar represents average of five simulations.
Figure 8: Proliferation capacity fixed at 10. Spontaneous death fixed at 0.05. Probability of stem cell and migration distance are varied to determine parameter settings which best replicate real growth patterns (determined by minimizing cost with respect to polynomial curve from Figure 5).

Figure 9: Data from Figure 8 plotted separately along each migration capacity. Curves fit to data are exponential.

Figure 10: Data from Figure 8 plotted separately along each probability of stem cell.
Figure 11: Visualization of simulated tumour and comparison of model growth curve versus target curve from experimental data (example case from Figure 8). $p_s = 0.02$, $\mu = 1$ ($\alpha = 0.05 \rho = 10$ held constant).

Figure 12: $p_s = 0.02$, $\mu = 3$ (increased migration distance compared to Figure 11)

Figure 13: $p_s = 0.06$, $\mu = 1$ (increased probability of stem cell compared to Figure 11).
Discussion

The first step was to establish a target curve. One would expect an exponential curve would provide a better fit for growth patterns, however, as Figure 5 indicates this is not the case. A cubic polynomial actually provides superior fit. There is an anomalous rise in this target curve (minimum is not centered at zero but rather closer to 10 days). This can be accounted for by the methodology of tumor injection. The cancer cells were suspended in a gel for injection, artificially increasing the initial tumour volume. This gel dissolves and it was common for the first few measurements to actually indicate a decrease in tumour volume.

Having a target curve, the six parameters were subsequently eliminated in pairs (a model is straightforwardly analyzed simultaneously in two dimensions, but it is much more difficult to make meaningful inferences in six dimensions). The first to parameters to be fixed were the number of iterations and lattice size. This decision was made from a computational standpoint. The computationally feasible scale of the in silico model is so many orders of magnitude smaller than in vivo tumours that number of iterations and lattice size are irrelevant; no matter what they are, they are not going to accurately reflect the in vivo tumour. However, this is acceptable because the information we are interested in is tumour growth pattern. We are building a stable model with useful medical applications, and not necessarily an exact replica.

Enderling et al. describe tumours as amalgamations of stem cell seeded “sub-tumours”. The tumour surrounding each stem cell, however, reaches steady state and is limited in size by proliferation capacity and spontaneous death (Figure 3). Thus it was hypothesized that growth is actually better conceptualized as discrete rather than continuous. Each new stem cell “steps” the overall tumour volume as the immediate volume surrounding it reaches steady state. Figure 7 indicates the search for reasonable “step” size. Since probability of stem cell is fixed at zero, the data represents a tumour seeded by a single stem cell; a single “step”. If the volume is too small then model unnecessarily sacrifices additional realism while if volume is too high growth will be very apparently discrete (the
“resolution” will be too low). Figure 7 indicates that proliferation capacity has the greatest effect on volume. A value of 10 is selected to be held constant in subsequent simulations. Probability of death has much less effect and a value of 0.05 is selected.

In Figure 8 the final two parameters, probability of stem cell and migration distance, are investigated. Figure 9 helps to visualize the effect of probability of stem cell. The fitted curves indicate similar exponential decrease in cost (more realistic growth patterns) regardless of migration distance. One problem with using cost as a comparison metric is that it does not intrinsically suggest a threshold under which values can be considered sufficient (there is not a natural point where to draw the line between a good cost and a bad cost). However, the graph indicates that all curves asymptotically approach zero around 0.04. Figure 10 helps to visualize the effect of migration distance, but no clear trend can be inferred.

Figures 11 through 14 serve to visualize the results of Figure 8 and provide some further qualitative information. Figures 11 and 12 have $p_s = 0.02$ while Figures 13 and 14 have $p_s = 0.06$. The improvement in growth pattern is very apparent from the graphs. Figures 13 and 14 differ in migration distance. While both follow realistic growth patterns, we can see that tumour morphology differs. We can qualitatively conclude that tumours with lower migration distances are rounder, while those with larger migration distances tend to be spindlier. It is thought that this is because stem cells in the spindlier tumour are able to travel farther before seeding a “sub-tumour”. However, this spindling behavior might be reduced if the model provided for a barrier in the form of surrounding healthy cells (as opposed to the free floating tumour).

An additional aspect of the model which could affect behavior is the handling of cell death. In the model dead cells simply disappear, while in reality dead cells would persist and affect growth of surrounding cells. Furthermore, although the visualizations indicate otherwise, it is possible for a tumour to migrate to the edge of the lattice (more likely to occur as migration distance increases). These edge effects would alter growth patterns as growth is limited to a reduced number of directions. One possible solution would be to increase lattice size, but again this increase computational demand.

Lastly, while it has already been explained that the model represents tumours of a significantly different scale than in vivo tumours, the end goal is to integrate the model with an angiogenesis model. The current angiogenesis model is approximately $\frac{1}{4}$ the size actual cancer size. Thus, while it is not necessary to match realistic scales, it will be necessary to match model scales.

**Conclusion**

Tumour volume of human breast cancer in mice was measured using ultrasound over the course of 50 days. The data was used to establish a target growth curve. The iterative tumour growth model presented by Enderling et al. was implemented in MATLAB. The model parameters, probability of stem cell, migration distance, proliferation capacity and probability of death, were tuned such that model growth patterns matched the target growth curve. An additional two parameters, number of iterations and lattice size, were introduced by the nature of *in silico* implementation. However, little investigation
was made into these parameters due to computational limitations and they were held constant. It was found that for a model consisting of a 100x100x100 lattice and undergoing 100 iterations, with proliferation capacity of 10 and probability of death 0.05, that probability of stem cell had the most significant effect on growth pattern. Increasing to $p_s=0.04$ saw cost decrease exponentially (indicating improvement in match the target curve) but additional increases saw negligible improvement. The effect of migration distance was found to be more qualitative in that it affects shape. Lower migration distances tend to produce rounder tumours while higher migration distances tend to produce spindlier tumours.
References


% %INITIALIZATIONS%%%
% model size & iterations (greatest direct effect on comp time)
% y = [0.01 10 0.05 1];
% n = 100;
% its = 500;
% model parameters
ps = y(1); % 0.01;
rho_max = y(2); % 10;
alpha = y(3); % 0.05;
mu = y(4); % 1;
% model "space"
rho = zeros(n,n,n);
rho(n/2,n/2,n/2) = inf;
% for indexing "space"
[Y,X,Z] = meshgrid(1:n,1:n,1:n);
all_x = reshape(X,n^3,1);
all_y = reshape(Y,n^3,1);
all_z = reshape(Z,n^3,1);
% for cell neighbourhood
open_nan = nan((2*mu+1)^3,1);
% for # of cells
volume = zeros(1,its+1);
tic;
for i1 = 1:its

% Migration
% reset indexing
x_cells = all_x;
y_cells = all_y;
z_cells = all_z;

% find cancer cell coordinates
x_cells = x_cells(rho > 0);
y_cells = y_cells(rho > 0);
z_cells = z_cells(rho > 0);

volume(i1) = length(x_cells); % assume volume directly proportional to # of cells

for cellID = 1:length(x_cells)

% clear x_open y_open
x_open = open_nan;
y_open = open_nan;
z_open = open_nan;

% dist
x_diff = x_cells(cellID) - mu;
if (i>0 && i<n+1) % boundary x

for i = (x_cells(cellID) - mu):1:(x_cells(cellID) + mu)

% end
end
end
for j = (y_cells(cellID) - mu):1:(y_cells(cellID) + mu)
    if (j>0 && j<n+1) %boundary y
        for k = (z_cells(cellID) - mu):1:(z_cells(cellID) + mu)
            if (k>0 && k<n+1) %boundary z
                if rho(i,j,k) == 0
                    c = c + 1;
                    x_open(c) = i;
                    y_open(c) = j;
                    z_open(c) = k;
                end
            end
        end
    end
end %end boundary x
end %end check open

x_open(1) = x_cells(cellID);
y_open(1) = y_cells(cellID);
z_open(1) = z_cells(cellID);

new_pos = cell(c*rand(1,1));
temp_rho = rho(x_cells(cellID), y_cells(cellID), z_cells(cellID));
rho(x_cells(cellID), y_cells(cellID), z_cells(cellID)) = 0;
rho(x_open(new_pos), y_open(new_pos), z_open(new_pos)) = temp_rho;
end

% Apoptosis
x_cells = all_x;
y_cells = all_y;
z_cells = all_z;

x_cells = x_cells((rho > 0) & (rho ~= inf));
y_cells = y_cells((rho > 0) & (rho ~= inf));
z_cells = z_cells((rho > 0) & (rho ~= inf));

survival = rand(1,length(x_cells));
for m1 = 1:length(x_cells)
    if survival(m1) <= alpha
        rho(x_cells(m1), y_cells(m1), z_cells(m1)) = 0;
    end
end

% Division
x_cells = all_x;
y_cells = all_y;
z_cells = all_z;

%output = output(input~=0);
x_cells = x_cells(rho > 0);
y_cells = y_cells(rho > 0);
z_cells = z_cells(rho > 0);

for cellID = 1:length(x_cells)
    %clear x_open y_open
    x_open = open_nan;
    y_open = open_nan;
    z_open = open_nan;
    c = 0;

    for i = (x_cells(cellID) - 1):1:(x_cells(cellID) + 1)
        if (i>0 && i<n+1) %boundary
            for j = (y_cells(cellID) - 1):1:(y_cells(cellID) + 1)
                if (j>0 && j<n+1)
                    for k = (z_cells(cellID) - 1):1:(z_cells(cellID) + 1)
                        if (k>0 && k<n+1)
                            if ~rho(i,j,k) %rho(i,j) == 0
                                c = c + 1;
                                x_open(c) = i;
                                y_open(c) = j;
                                z_open(c) = k;
                            end
                        end
                    end
                end
            end
        end
    end
end %end boundary x
end %end check open

if c > 0
    new_pos= ceil(length(find(isnan(x_open)==0))*rand(1,1));
    if isinf(rho(x_cells(cellID),y_cells(cellID),z_cells(cellID)))
        new_type = rand(1,1);
        if new_type <= ps
            rho(x_open(new_pos),y_open(new_pos),z_open(new_pos)) = inf;
        else
            rho(x_open(new_pos),y_open(new_pos),z_open(new_pos)) = rho_max;
        end
    else
        rho(x_open(new_pos),y_open(new_pos),z_open(new_pos)) = rho(x_cells(cellID),y_cells(cellID),z_cells(cellID)) - 1;
        rho(x_open(new_pos),y_open(new_pos),z_open(new_pos)) = rho(x_cells(cellID),y_cells(cellID),z_cells(cellID));
    end
end

scatter3(x_cells,y_cells,z_cells);
axis([0 n 0 n 0 n])
title({i1;volume(i1)});
drawnow
end
t = toc;

%adding final point
x_cells = all_x;
x_cells = x_cells(rho > 0);
volume(length(volume)) = length(x_cells);
volume = volume/max(volume);

figure
plot(0:(50/its):50, volume/(max(volume)))
hold on
day = 0.5:0.5:50.5;
target = 0.000005*day.^3 + 0.0002*day.^2-0.008*day + 0.0469;
plot(day, target);
%cost = sum(sum(volume-target).^2)/sum(sum(target.^2))
Appendix II

day = 0.5:0.5:50.5;
target = 0.000005*day.^3 + 0.0002*day.^2 - 0.008*day + 0.0469;

n = 100;
its = 100;

tempvs = zeros(3,its+1);

cost = zeros(4,1);
tempcost = zeros(1,3);

for i = 1:4
    for j = 3
        for k = 1:5
            [tempvs(k,:), ~] = TGTest([0.02*i, 10, 0.05, j], n, its);
            tempvs(k,:) = tempvs(k,:)/max(tempvs(k,:));
            tempcost(k) = sum(tempvs(k,:)-target).^2/sum(target.^2);
        end
        cost(i,j) = mean(tempcost);
    end
end
Appendix III

volumes = zeros(5,5);
ts = zeros(5,5);
n = 100;
its = 200;

temps = zeros(1, its+1);

for i = 0:4
    for j = 0:4
        [temps, ts(i+1,j+1)] = TGTest([0, i*5, j*0.025, 2], n, its);
        volumes(i+1,j+1) = mean(temps(50:end));
    end
end