Kidney Morphogenesis Cellular Automaton - Model Details

Ben Lambert

May 23, 2014

1 Introduction

The aim of this model is to recapitulate the development of the Uretic Bud *in vivo* in 2D. However, it is also hoped that the model may be useful for the study of *in vitro* explants. The details of the model are described in brief below:

- A 2 dimensional rectangular domain, where the points in the array represent the presence of:
  - Epithelium cells - which consume GDNF
  - Metanephric mesenchymal cells (MM) - which produce GDNF
  - Extracellular matrix (ECM) - which allow for free diffusion of GDNF
- Epithelium cells comprising either:
  - A *flat* membrane at the base of the simulation domain (aimed at recapitulating the *in vivo* conditions from which the UB initially forms *in utero*.)
  - A mass of epithelium cells with in general, curved boundaries, suspended towards the centre of the simulation domain (representative of the *in vitro* explant experiment initial conditions.)
- A diffuse collection of mesenchymal cells initially separate from the epithelium.

2 GDNF field

2.1 Steady state reaction-diffusion equation

Cells in the MM produce GDNF, and it diffuses freely across the ECM layer, and through the epithelium (being consumed by the latter). The reaction-diffusion type equation here will be assumed to be in equilibrium, since the diffusive timescale of GDNF is much less than that of the timescale of cell division. As such, the following is the form of the equation being solved:

$$D_G \nabla^2 = \Phi_G \tag{1}$$

Where in (1) D_G refers to the diffusion coefficient for GDNF, and Φ_G is the local rate of GDNF production or consumption (dependent on whether the cell in question is epithelium or mesenchyme). Specifically the rate of GDNF consumption is assumed to have the following form:

$$\Phi_G = \begin{cases} K_G G, & \text{for epithelium.} \\ -\rho_G, & \text{for mesenchyme.} \\ 0, & \text{for the extracellular matrix.} \end{cases} \tag{2}$$

It is assumed that the rate of GDNF consumption is linearly-dependent on the concentration of substrate. This assumption is likely violated when the GDNF concentration is high, and the cell Ret-receptors are saturated. In later versions of this model, it may be better to assume a Hill-type reaction rate. (Although it is unclear as to what range of concentrations of GDNF are likely to saturate the Ret receptors, and whether these are likely to be encountered either *in vivo* or in experiments conducted thus far *in vitro*.) However, this sort of non-linear consumption function is best handled by solving the parabolic time-dependent reaction-diffusion equation (see section 2.2). It is assumed to begin with that the mesenchyme GDNF production rate is constant, partly for simplicity, although this will be relaxed in latter models which, for example, allow mesenchymal GDNF production rates to be a positive function of their proximity to epithelium cells (perhaps as a proxy for Wnt11 levels).

The equation in (1) is non-dimensionalised using the following transformations:

$$\eta = \frac{x}{\Delta} \tag{3}$$

$$g = \frac{G}{G_x} \tag{4}$$

Where in (3) Δ refers to the typical cell dimensions (approximated as 5 μm),and G_x is the concentration of GDNF typically found *in vivo* (a value is currently not assigned here, since it is not strictly necessary for the non-dimensionalised simulation). These transformations result in the following non-dimensional form of the steady-state reaction-diffusion equation:

$$\nabla_\eta^2 g = \frac{\phi_g}{d_g} \tag{5}$$

Where in (5), $d_g = \frac{D_G}{K_G \Delta^2}$, $\phi_g = \frac{\Phi_G}{K_G G_x}$, and ∇_η^2 is the laplacian in the non-dimensional η spatial coordinates.

The boundary conditions which are assumed are: no-flux at the base of the Wolffian Duct and top of the simulation domain, and periodic boundary conditions at either width. A finite difference approximation is used, with the no-flux boundary conditions at the top and bottom of the domain represented by the following relations for the non-dimensional GDNF field:

$$g_{0,j} = g_{1,j}$$
$$g_{M,j} = g_{M+1,j}$$

where this holds $\forall j = 1, ..., N$. M here is the depth of the simulation domain in terms of cell dimensions. Similarly, for the periodic boundary conditions:

$$g_{i,0} = g_{i,N}$$
$$g_{i,N+1} = g_{i,1}$$

where this holds $\forall i = 1, ..., M$. N here is the width of the simulation domain in terms of cell dimensions.

The finite-difference approximation for solving the steady-state reaction-diffusion equation is hence of the form:

$$g_{i+1,j} + g_{i-1,j} + g_{i,j+1} + g_{i,j-1} - (4 + \epsilon_{i,j})g_{i,j} = -\psi_{i,j} \tag{6}$$

Where in (6):

$$\epsilon_{i,j} = \begin{cases} \frac{1}{d_g}, & \text{for epithelium.} \\ 0, & \text{for mesenchyme.} \\ 0, & \text{for the extracellular matrix.} \end{cases} \tag{7}$$

Also, in (6):

$$\psi_{i,j} = \begin{cases} 0, & \text{for epithelium.} \\ \frac{\gamma}{d_g}, & \text{for mesenchyme.} \\ 0, & \text{for the extracellular matrix.} \end{cases} \tag{8}$$

Where in (8), $\gamma = \frac{\rho_G}{K_G G_x}$ captures the relative rate of GDNF production by the mesenchyme compared to the rate of GDNF consumption by epithelium under normal cellular conditions. It is assumed that $\gamma = 1$ in the simulations, in the absence of information regarding the relative weight of these two mechanisms. The diffusion rate constant, D_G, is assumed to be 10 $\mu m^2 s^{-1}$ (following Menshykau and Iber). If we assume that the diffusive time scale is similar to that of the consumption rate of GDNF we find that:

$$k_G \sim \frac{D_G}{\Delta^2_{domain}} \tag{9}$$

Where in (9), Δ_{domain} refers to the simulation domain spacial dimensions. Typically, the simulation takes place in a domain with a width of 100 cells and a similar depth. This leads to an effective reaction rate constant of $k_G = 1 \times 10^{-5} s^{-1}$. This results in a dimensionless diffusion rate of $d_g = 4000$. Previously, without this approximation, the simulation was conducted using a value of $d_G = 200$; resulting in heterogeneous GDNF concentrations across the domain, with relatively low values in epithelial crypts. With this higher value of the effective diffusion rate however, the concentration is much more uniform, with low concentration gradients resulting. As a result, this higher value of the diffusion rate made it considerably harder to generate branching patterns.

In the simulation, it is assumed that the GDNF field does not vary significantly within a particular time step. As such (in order to improve computational speed), the GDNF field is only updated at the end of each time step, not after each cell movement or proliferation. Ultimately however, it will be important to test whether the model conclusions are sensitive to this assumption.

2.2 Time-dependent reaction-diffusion numerical approximation

This section specifies the details of the non-dimensionalisation of the reaction-diffusion equation to be used to study branching *in vitro* of explanted kidney epithelium cells in a GDNF broth. Another reason for using a parabolic quasi-steady state reaction-diffusion equation is that it allows for easier inclusion of non-linear GDNF consumption/production functions than that for the elliptic case. Both explicit and implicit finite difference schemes are employed to numerically simulate the system.

2.2.1 Details of the non-dimensionalisation

The specific reaction-diffusion equation relevant to the epithelium-broth system is of the form:

$$\frac{\partial G}{\partial t} = D_G \nabla^2 - \Phi_G \tag{10}$$

Where in (10) D_G refers to the diffusion coefficient for GDNF, and Φ_G is the local rate of GDNF consumption which occurs at epithelium cells.

The equation in (10) is non-dimensionalised using the following transformations:

$$\eta = \frac{x}{\Delta} \tag{11}$$

$$g = \frac{G}{G_x} \tag{12}$$

$$\tau = \frac{t}{T} \tag{13}$$

Where in (11) Δ refers to the typical cell dimensions (approximated as 5 μm),and in (12), G_x is the concentration of GDNF typically found *in vivo* (a value is currently not assigned here, since it is not strictly necessary for the non-dimensionalised simulation). In (13), T is the time scale of a typical cell division; assumed to be $T \sim 1hr$. These transformations result in the following non-dimensional form of the time-dependent reaction-diffusion equation:

$$\frac{\partial g}{\partial \tau} = d_g \nabla_\eta^2 g - \phi_g \tag{14}$$

Where in (14), $g = G/G_x$, $d_g = \frac{TD_G}{\Delta^2}$, and, $\phi_g = \frac{\Phi_G}{K_G G_x}$. Note here that, the dimensionless diffusive rate here represents the ratio of two time scales: the cell division time as compared to the diffusion time scale. The latter can be shown to be (for a diffusion rate of $D_G = 10\mu m^2 s^{-1}$, and a cell dimension of $\Delta \sim 5\mu m$) approximately of order $\frac{\Delta^2}{D_G} \sim 2.5s$; resulting in an dimensionless diffusion rate of order $d_g \sim 1500$.

The two ways considered to numerically simulate the system were *explicit* and *implicit* finite difference schemes. They numerically approximate (14), by using the following schemes:

$$\mathbf{g^{t+1}} = \Delta_\tau[(d_g \nabla_\eta^2 + \frac{I_{MN}}{\Delta_\tau})\mathbf{g^t} - \mathbf{\Psi}(\mathbf{g^t})] \tag{15}$$

$$\mathbf{g^{t+1}} = [I_{MN} - \Delta_\tau d_g \nabla^2]^{-1}(\mathbf{g^t} - \Delta_\tau \mathbf{\Psi}(\mathbf{g^t})) \tag{16}$$

Where (15) refers to the *explicit* scheme, and (16), to the *implicit* one, and Δ_τ is the time step taken. In (15) and (16), $\mathbf{\Psi}$ is a vector which contains all GDNF consumption and production terms, and can, in general, be a non-linear function of GDNF. It is assumed that $\mathbf{\Psi}$ is formed using the cell positions corresponding to the current (known) time period. Although this results in a somewhat 'bastardised' implicit scheme, it is perhaps a reasonable approximation, as cell positions are not likely to change significantly between the time steps taken, since the cell cycle is much longer that the typical GDNF consumption time. In (16) the RHS will be calculated using Matlab's inbuilt backslash operator to avoid the computational overhead of calculating the matrix inverse.

3 Epithelium cell behaviour

3.1 Algorithm governing cellular behaviour

In this section I describe the behaviours of the epithelium cells in the cellular automaton model. In the simulation, each epithelium cell is visited in a random order, (actually epithelium and mesenchyme are updated together, so the randomised order corresponds to both cell types), and updated at each discrete time step according to the following pseudo-algorithm:

- *Available cells* - Are neighbouring cells available for movement or proliferation? If yes, proceed to step 2. If no, move on to next cell. Whether a cell is available depends on the specific rules in place (see section 3.2 for more details).

- *Move or proliferate* - Draw a random number $X \sim unif(0,1)$. If $P_{move} > X$ then proceed to *move*. If not, proceed to *proliferate*.

- *Move*

  1. *Calculate the probability that a move takes place, P_m* - Based on the specific rules in place calculate the probability that a move takes place (see section 3.3 for more details).
  2. *Does a move take place?* - Draw a random number $X \sim unif(0,1)$. If $P_m > X$ then proceed to next step. Otherwise, consider next epithelium cell.
  3. *Calculate probability of moving to each of the available cells* - Based on the specific rules being used, calculate probabilities of moving to each of the available target cells (see section 3.4 for more details about the options for rules used here).
  4. *Choose amongst the available cells in accordance to their probability and move the cell in question.*

- *Proliferate*

  – The framework is exactly analogous to that of *Move* detailed above. However, the rules chosen for proliferation or moving are allowed to be different in the simulation. In proliferation, the mechanism is slightly different due to the fact that a daughter cell is created in the target cell, rather than a cell simply moved there.

- Consider next cell, returning to first step.

The program has been created in Matlab which allows for a number of permutations of the rules to be tried by the user. However, there are some combinations which are not allowed due to internal inconsistency amongst them. The idea is to allow the model's results to be easily tested for robustness to assumptions made.

3.2 Rules governing whether neighbouring cells are available

In this section I will detail the various different rules which govern into which cells an active cell (the one being updated) can move or proliferate. These rules are allowed to be different for cells which are moving or proliferating.

1. *Number of neighbours* - the choice here governs whether 4 (up, down, left, and right) or 8 (all points of a compass) are surveyed as potential candidates for a move or proliferation.
2. *Connectivity of cells*

   (a) All vacant cells are allowed.

   (b) Moves or proliferations are allowed into vacant cells only if the active cell (the one being moved, or the daughter cell being created) are connected to other cells. Here connectivity means that there is at least one 4-neighbour (up, down, left or right) which is occupied by an epithelium cell. For a visual explanation of this, see figure 1.

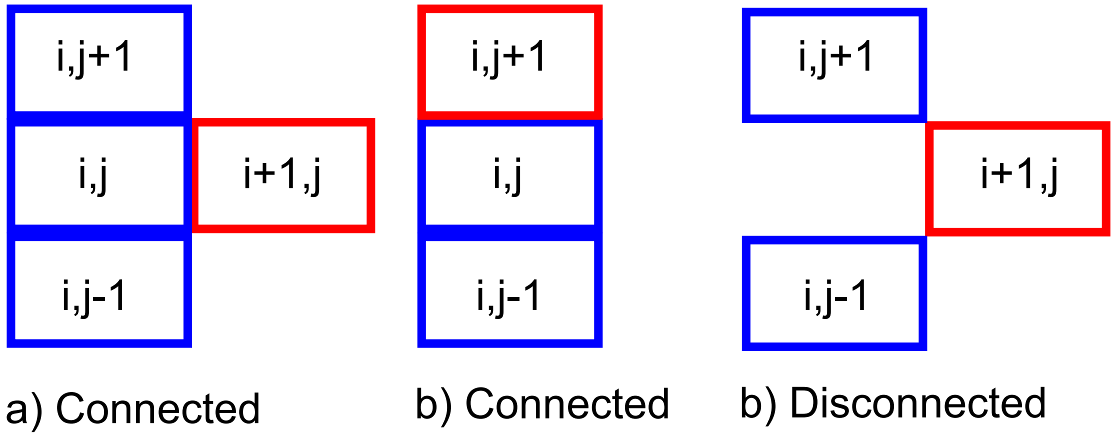

Figure 1: An explanation of 'connectivity' for epithelium cells. All cells here are epithelium, the one highlighted in red is the cell which has been moved. The first two moves would be allowed, whereas the last is not.

(c) Same as above, but more stringent. Only allow moves if all cells are connected. This allows for the possiblity of a move leaving one cell unconnected.

(d) All vacant cells and mesenchyme are allowed as possible move/proliferation locations. If a movement into a cell occupied by a mesenchymal cell occurs, then the mesenchyme cell is removed from the simulation.

(e) Same as 2d, although only allow movement into a space if that active cell remains connected.

(f) Same as 2e, although instead of killing mesenchyme cell (upon movement into its space), move the mesenchyme into a vacant space neighbouring it (as shown in figure 2). If there are no vacant spaces available for the mesenchyme cell to move into, then the move is not allowed. If multiple spaces are available for the mesenchyme to move into, then a space is chosen either at random or weighted towards the direction in which they were pushed (see section 4.1).

(g) Same as 2f, but prevent any epithelium movements if they lead to a mesenchyme becoming 'engulfed' by surrounding epithelium (see section 4.2 for a further discussion).

(h) Allow movements of epithelium into both vacant and mesenchyme grid locations (assuming the active cell is connected). However, allow mesenchyme to be moved non-locally (by non-local here I mean to a location which is not neighbouring) to a vacant location within a rectangular domain defined by the direction in which it was pushed. See section 4.3 for more details.

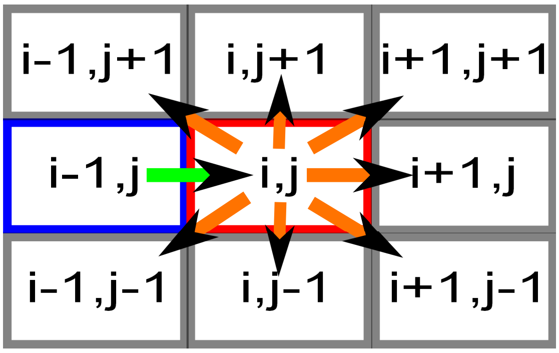

Figure 2: The 7 available moves for a mesenchyme at (i, j) after a epithelium at $(i-1, j)$ moves into its cell location. The grey cells are assumed to be vacant. Note that this epithelium move would not be allowed under certain rules as the active cell would become unconnected.

The above rules 2b, 2c, 2e, 2f, 2g and 2h are phenomenlogical, but are aimed at mimicking the cell-cell adhesion experienced by the epithelium. Rules 2d, 2e, 2f, 2g, and 2h are included as to allow an interaction between the epithelium cells and the mesenchyme, when the former move into the area occupied by the latter. It is thought that 2f is the most biologically realistic, since excess cell death is not typically observed, and cells do not 'jump' non-locally. However, there are issues with this particular framework, since it does not accommodate the ability of an incident epithelium layer to push on a dense layer of mesenchyme - an interaction which is observed in practice. This type of behaviour is not observed since there may not be any vacant spots for mesenchyme cells to move into in this single occupancy CA model. This motivated the introduction of non-local jumping of mesenchyme in schema 2h which is described in detail in section 4.3. Rule 2g was introduced to investigate mechanisms which would prevent mesenchyme cells becoming isolated within an epithelium mass; which had occurred with the previous rules. A description of what is meant by 'engulfment' and the mechanism of operation for this rule are detailed in 4.2.

3.3 Rules governing the calculation of P_m or P_p - the probability of a move or proliferation occurring after the action has been chosen

In this section I describe the available choices of model rules which govern the calculation of the probability of a move/proliferation occurring takes place. To be clear, this is the step after the choice has been made to either go down the algorithmic branches corresponding to *Move* or *Proliferate* respectively. It is chosen to make the decision as to whether to continue along each branch here, rather than at the stage before to allow for potentially different rules governing moving and proliferating respectively. The rules available for calculating whether or not a move/daughter cell creation takes place are given below:

1. The probability of a move/proliferation is a constant specified by the user.
2. The probability of a move/proliferation is a positive function of the local GDNF concentration (the level of GDNF for that cell at that grid point). The equation used here to specify the probability is that of a probit model

$$P(action) = \Phi(c_1 + c_2 g) \tag{17}$$

   Where in (17), Φ is the standard normal CDF, and *action* can either be a move or a proliferation event.

3. Move/proliferation probability is determined by the sum of all local positive GDNF gradients, between each of the available neighbours and the current location. Again, the model used here for the probability is the probit model:

$$P(action) = \Phi(c_1 + c_2 \sum_{allowed} (g^{neighbours} - g^{current}) \times \mathbb{1}(g^{neighbours} > g^{current}) \tag{18}$$

   Where in (18), the indicator function $\mathbb{1}(g^{neighbours} > g^{current})$, is equal to one if the gradient in GDNF allowed by the move is positive, and zero otherwise.

Rule 2 is aimed at allowing more cellular activity (moves or proliferations) in areas where GDNF concentration is higher. The use of a normal CDF here is partly to constrain the probability to lie in the appropriate (0,1) range, but can also act to provide thresholding-type behaviour, which may be necessary to generate discrete bud formation from the Wolffian Duct *in vivo* opposed to a general 'swelling'. Rule 3 allows more cellular activity in areas where there is more GDNF to be gained by a potential move or proliferation. I am less sure as to the biological realism of this latter rule, although for it could be thought of as a type of chemotaxis for moving cells, or orientated cell division for proliferating cells.

3.4 Rules governing selection of move target or daughter cell location

This last section on rules details how to determine the location (amongst the available, allowed neighbouring cells) into which either a cell should move or create a daughter cell. The rules governing how the probabilities of choosing a given target location are given below:

1. The probabilities of choosing each of the possible target cell are equal, and given by:

$$P(target) = \frac{1}{N_{target}} \tag{19}$$

   Where in (19), N_{target} specifies the total number of allowed possible target cells.

2. The weights given to a specific target cell are given by:

$$W(target) = \Phi(c_3 + c_4(g^{target} - g^{current})) \tag{20}$$

   The weights across all targets are then used as weights in a Dirichlet distribution to calculate probabilities which sum to 1; with the higher weights corresponding to higher probabilities.

3. The weights given to a specific target cell are related to the percentage changes in GDNF:

$$W(target) = \Phi(c_3 + c_4\frac{g^{target} - g^{current}}{g^{current}}) \tag{21}$$

   The weights across all targets are then used as weights in a Dirichlet distribution to calculate probabilities which sum to 1; with the higher weights corresponding to higher probabilities.

4. The probabilities of choosing a specific target cell are given by a multinomial logistic distribution:

$$P(target_n) = \frac{exp(c_3 + c_4(g^{target} - g^{current}))}{\sum\limits_{alltargets} exp(c_3 + c_4(g^{target} - g^{current}))} \tag{22}$$

   Where in (22), the benefit of this particular form is that the probabilities naturally sum to 1.

Rules 2, 3, and 4 represent chemotaxis (when considering movement), and orientated cell division (when considering proliferation). It is not clear to me whether orientated cell division can occur along concentration gradients, but this is provided for model completeness.

After conducting the simulations, it does not appear that there are significant differences between these different methods. In terms of those rules which weight towards those target cells which have the highest (positive) gradient in GDNF, the preferred rule is 4, which naturally provides a way of creating probabilities which sum to 1 overall.

4 Mesenchyme cell behaviour

The current version of the simulation does not allow for 'active' behaviour of the mesenchyme. Their only movement behaviour is their ability to be either 'killed' or moved out of the way, by advancing epithelium cells. Later versions of the simulation will allow for autonomous mesenchymal behaviour; for example attraction towards epithelium (or Wnt11 released by the epithelium), or other mesenchyme cells. In this section the rules which dictate how mesenchyme move in reaction to encroaching mesenchyme are explained.

4.1 Local mesenchyme movement

Under certain rules which govern the *allowed* movements of epithelium, a move or proliferation is allowed into a cell location which is currently occupied by a mesenchyme cell; assuming there are neighbouring (local) vacant cells into which the latter can move. A simple rule that can be used here is to allow mesenchyme cells to move into each of the available locations with a probability that is uniform across all moves. This type of movement is likely a simplification that is not physically realistic as cells are more likely to be move in a direction in which they are pushed. To allow for a bias towards local movement in a direction in which cells are pushed, a mechanism was created which was aimed at better representing the physical interaction between mesenchyme and epithelium. It is the details of this mechanism which are described in this section.

For every move by an epithelium from a position (i, j) to a neighbouring position (i', j') occupied by a mesenchyme it is possible to define a vector representing that movement:

$$\Delta(i, j)^e = (i', j') - (i, j) = (\Delta i^e, \Delta j^e) \tag{23}$$

This vector is restricted either to be one of the following unit vectors (they are unit in the simulation domain as the distances are non-euclidean): $(0, 1), (1, 0), (0, -1), (-1, 0), (1, -1), (1, 1), (-1, -1), (-1, 1)$. Where the first four are the only set of moves available for the case of considering only four nearest neighbours, whereas the whole set are available for the 8-NN case. The set of available spaces into which the mesenchyme can move can then be found, and associated with each is a movement vector. For example a movement of a mesenchyme from (i, j) to $(i, j + 1)$ is $(0, 1)$. In (23) the vector $\Delta(i, j)^e$ is then compared to the set of all available movement vectors by using a Least-Squares measure of difference:

$$S^k = (\Delta i^e - \Delta i^k)^2 + (\Delta j^e - \Delta j^k)^2 \tag{24}$$

Where in (24) the index k indicates a particular available move allowed for the mesenchyme cell. Rather than disallowing moves with relatively larger values of S^k the eventual move was chosen probabilistically by using a multinomial logistic function to specify the individual probabilities:

$$P(move_k) = \frac{exp(-\beta S^k)}{\sum_k exp(-\beta S^k)} \tag{25}$$

Where in (25), β is a positive parameter which measures the strength of discrimination against moves which are not in the same direction as the moving epithelium. A visual description of this rule is shown in figure 3, for the case of $\beta = 0$ against $\beta >> 0$.

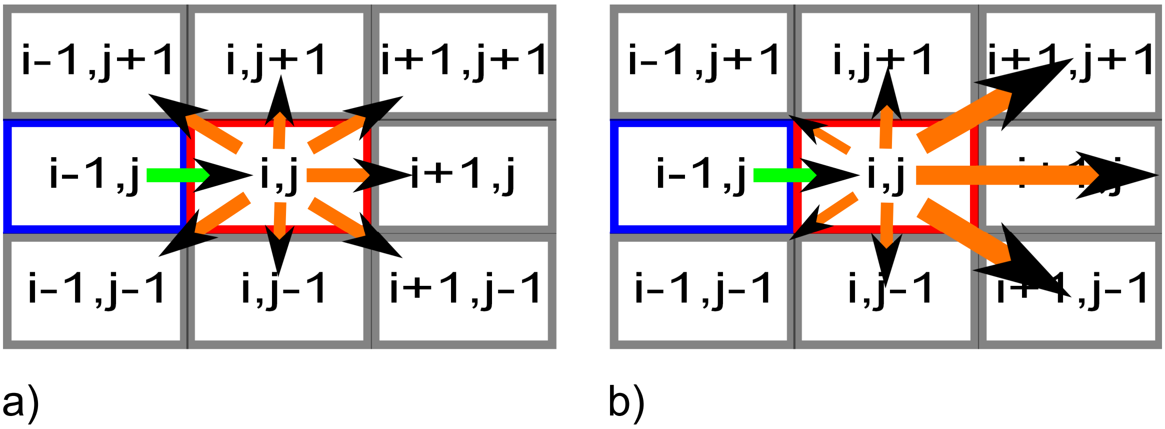

Figure 3: The available moves for a mesenchyme at (i, j) after a epithelium at $(i-1, j)$ moves into its cell location. The probability that a move occurs is given by the length of the (orange) arrows. In a) the strength of discrimination against moves which are not aligned with the epithelium move is zero. In b) there is a high degree of discrimination against vectors of movement which are at angles to the movement vector of the epithelium.

4.2 Mesenchyme engulfment

Whereas in this cellular automaton it is possible for mesenchyme cells to become surrounded by epithelium, this is not observed in reality. In this section a mechanism is described which was implemented to try to reduce the frequency of this biologically-unrealistic occurrence in simulations.

'Engulfment' of a cell is the term I use to describe a situation where a mesenchyme is surrounded by a high number of epithelium cells. The number of surrounding epithelium cells above which a mesenchyme is deemed 'engulfed' is a parameter which I allow to be varied from a maximum of 8 (no possible engulfment), to zero (no nearest neighbours can be epithelium.)

There were two methods which were used to reduce the frequency of 'engulfment' of mesenchymes:

1. Disallow movement of mesenchyme cells to positions where they would be 'engulfed'.
2. Disallow movements of epithelium into cells which would result in situations where adjoining mesenchyme become engulfed (either directly or indirectly if a mesenchyme must be moved out of the way into a situation whereby it is surrounded by epithelium.)

The implementation of this method (dependent on the threshold number of neighbouring epithelium cells which is chosen) could lead to a reduction in the occurrence of mesenchymal engulfment, although the magnitude of this decrease in frequency it should be said is not always visually evident. However, if a low threshold was applied (only a small number of neighbouring epithelium required before declaring a mesenchyme engulfed), biologically-infeasible patterns arose, where individual mesenchyme were surrounded by a layer of vacant space.

In summary I do not think this method is a particularly good choice for a biologically-realistic model of a UB which does not engulf mesenchyme cells. I also do not think that the rules which are described are reasonable as they assume a degree of sentience for the mesenchymes (where they 'look' to see if a move is allowed before making it), which is unrealistic.

4.3 Non-local movement of mesenchyme

It is observed that epithelium cells in the tip of a UB interact with an entire 'mass' of mesenchyme; eventually (via mutual interactions) resulting in a number of distinct mesenchyme subpopulations. With single occupancy of sites in a cellular automaton model, it is difficult to imagine how this type of behaviour could be recapitulated with local movement of mesenchymes out of the way of approaching epithelium.

A possible solution to this would be to allow for multiple occupancy of sites by mesenchymal cells, although with multi-occupancy sites preferring to reduce their occupancy (by moving mesenchyme cells towards less densely-populated areas). This would require considerable work to accommodate these changes in the current model framework. It was suggested that this type of aggregate behaviour (where epithelium masses push on dense groups of mesenchyme) could be attained by allowing mesenchyme cells to be moved non-locally out of the way of encroaching epithelium. Although, this behaviour is less realistic than local movements, (as cells do not simply jump position), it can be thought of as representing the process whereby displacement of mesenchyme occurs via contact with epithelium, causing a movement of the entire mass. Since the mesenchyme cells are not heterogeneous in terms of their properties, in terms of the model, the movement of an individual cell non-locally to a position which is vacant is equivalent to a shuffling of the mesenchyme mass which results in an 'outer' mesenchyme occupying the same vacant position (assuming all other positions are the same between the two.) Furthermore, since the process causing the movement of the mass of mesenchyme is likely mechanical, its time scale can be thought of as faster than the general movement of cells; lending some justification for non-local jumping as representing movements which occur at a faster rate than cell movement or proliferation.

A schema which allows for non-local movement of mesenchyme cells is presented here. In defining which space of cells were available to a mesenchyme after it being encroached on by a epithelium cell, it is necessary to introduce the following concepts:

1. *Principal axis* - the axis which is parallel to the direction of the movement vector of the epithelium as described in (23).
2. *Secondary axis* - the axis which is 'perpendicular' to the direction of the movement vector of epithelium.

These two concepts are explained visually in figure 4. In defining these axes a rectangular space is traced out according

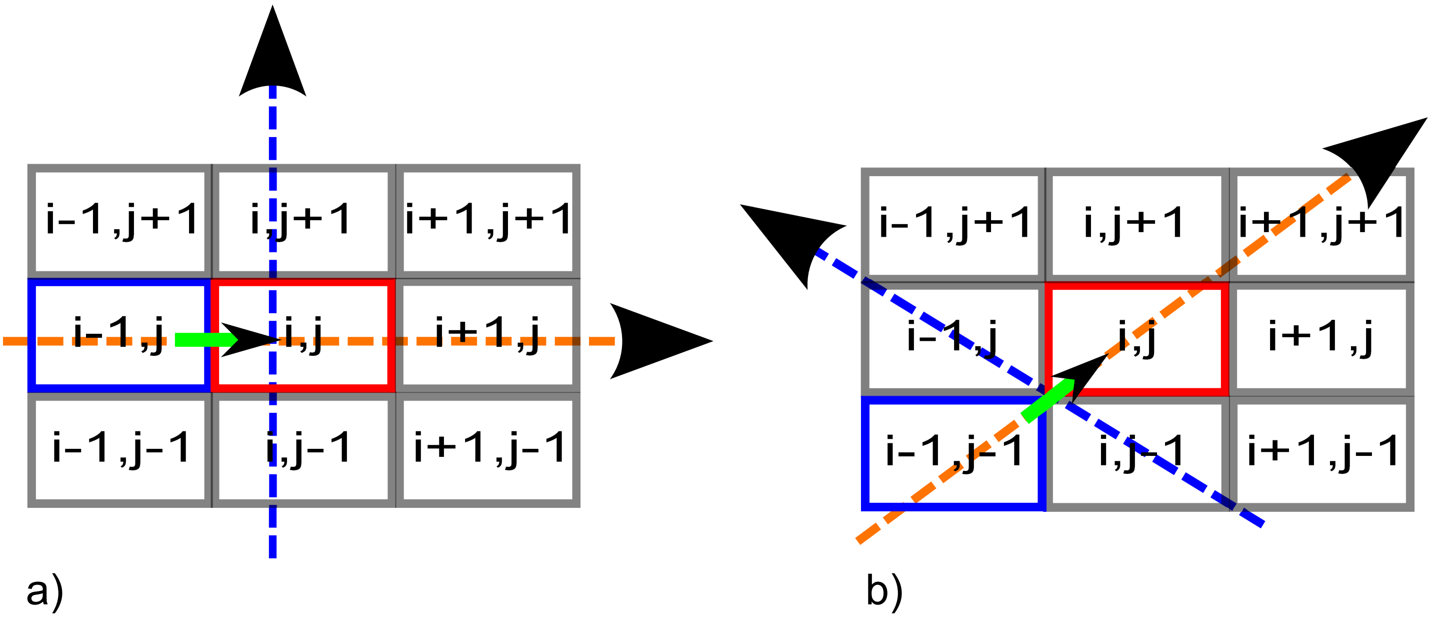

Figure 4: The principal axis (in orange) and secondary axis (in blue) for two example movements. In a) the movement vector is horizontal, whereas in b) it is diagonal. The axes are not visually perpendicular in b) due to the fact that the cells are represented as rectangles rather than squares.

to the maximum number of moves along the principal axis which are allowed, as well as the maximum number of moves sideways relative to the motion vector for the epithelium. The length of the rectangular space is chosen to start one unit behind (along the principal axis) the mesenchyme being considered, and ends after the maximum move length has been reached, p; resulting in a length of $p+2$. The width of the domain is given by specifying the maximum number of moves along the secondary axis which were allowed, s; resulting in a width of $2 \times s + 1$. This allowed movement domain is shown in figure 5.

Currently, there is no bias towards any of the cells which lie in the rectangular domain of allowed moves. In reality it is more likely that moves will occur if they are at a small angle to the principal axis, and are closer to the current cell location. The former is dealt with heuristically by specifying a distance $s << p$ to ensure that the movements of cells are, by majority at a small angle to the principal axis. The latter could potentially be dealt with by specifying a cost which depends on distance from the current cell. It would also be possible to allow a wider range of angular movements, with those movements resulting in the largest angle between the principal axis and the movement vector being discriminated against most strongly. However, both of these extra movement constraints require more parameters, and make the model more complex, with the possibility that they may not add much insight to the situation at hand. These may be revisited in the future however.

5 Cell death

The current version of the simulation does not allow cell death directly; only indirectly through the epithelium's ability to (dependent on the exact rules implemented) 'overtake' a position currently occupied by the mesenchyme. Later versions will allow the user to stipulate differing death rates for the epithelium and mesenchyme, and also allow these to be context-dependent (dependent on, for example, the connectivity of the cell in question).

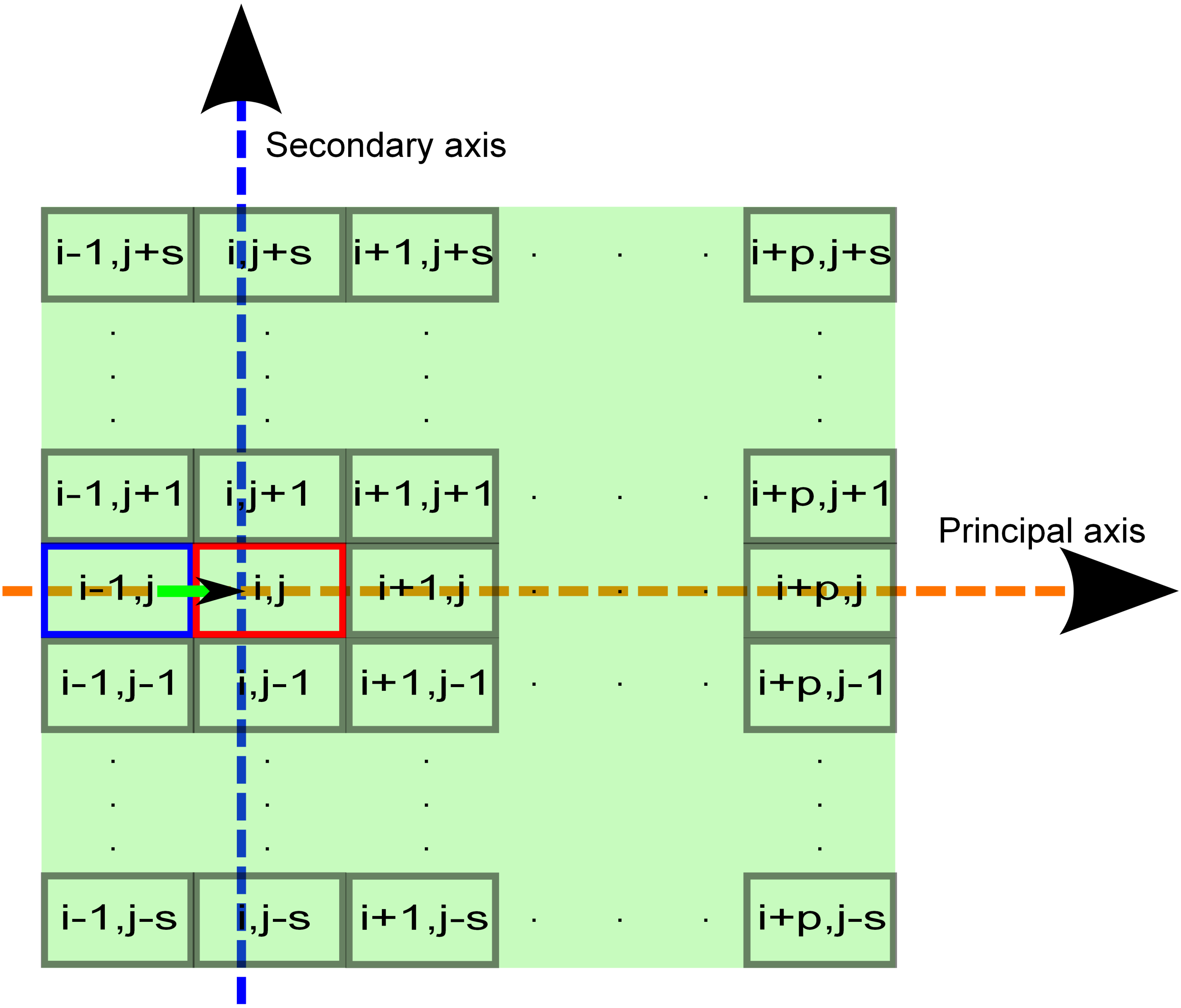

Figure 5: The space (coloured green) of available moves for a mesenchyme at (i, j) after a epithelium at $(i - 1, j)$ moves into its cell location if non-local cell movement is allowed. Here p and s refer to the maximum allowed move distance forward and sideways respectively.

6 Creation of the epithelium layer

Two different schema are considered for the creation of an epithelium layer:

- A flat layer of epithelium at the base of the simulation domain to emulate *in vivo* formation of the initial Uretic Bud. Although this layer is *in rerum natura* mono-layered, I choose to use a multi-layered epithelium. This is to allow for the fact that the simulation domain is two dimensional whereas in practice it is three dimensional. Making the layer multi-layered allows cells which are outside of the simulation plane to pass into the layer bordering on the ECM/mesenchyme.

- A randomised mass of epithelium cells towards middle of the simulation domain representing *in vitro* experimental conditions. The idea here is that branching arises partly as a result of local depletion of GDNF concentration in 'crypts' of epithelium next to outgrowths.

7 Branching measurement and visualisation

7.1 Perimeter-to-area ratio vs circle

A circle is the 2 dimensional object which has the lowest perimeter-to-area ratio (or put another way, it maximises area for a given perimeter). An object which is highly branched will, by contrast, have a relatively high ratio of perimeter-to-area. Hence, by comparing the perimeter-to-area ratio from the mass of epithelium with that of a circle with the same number of cells (the measure of area in the CA), the latter having a ratio of $ratio = \frac{2}{r}$, this could be taken to be a reasonable measure of the level of branching in the epithelium mass. However, it should be noted that this isn't necessarily a measure of branching, and could just be a measure of the fractal nature of the mass, without *per se* having distinct branches.

It is difficult to exactly calculate the perimeter of a contiguous mass of epithelium cells in this cellular automaton model, since this would involve coding the specific define characteristics of a cell on the edge of the mass. Instead of calculating the perimeter exactly, I borrow from medical imaging segmentation, using the Matlab function *bwconncomp* to calculate an approximate measure of the perimeter. This same functionality also yields an approximate area, which was compared to the exact area (simply the number of epithelium cells), and found to be in good correspondence. Visually the approximate perimeter also appeared to follow that seen, but further work here is needed to confirm that the approximate method here is reasonable (note to Ben: an idea - calculate the gradient of the image in both x and y and threshold it. All the perimeter cells should be connected. Then just use bwconncomp on the thresholded image. You can use the latter to calculate get the pixel ids.)

7.2 Medial axis skeleton branching

Another method to investigate branching in topology is to find the medial axis skeleton of the closed curve representing the perimeter of the object. The medial axis (in two dimensions) is the locus of points within the object which have two (or more) nearest points on the edge of the image. This means that a circle drawn with its centre at a particular point on the locus, will be tangent to the perimeter contour at two or more places. Example of medial axis skeletons are shown in figure 6 for an ellipse and an arbitrary perimeter.

Luckily Matlab has already a built-in function, *bwmorph*, which estimates the medial axis skeleton of an object, and also allows the user to extract the number of implied branch points from the skeleton. The visualisation of the estimated medial axis skeleton along with the number of branch points appears to be a nice way to estimate the degree of branching occurring within the epithelium mass. The latter was normalised by the area of the figure, as this appeared to be a better method of discriminating between whether an object was spherical vs branched.

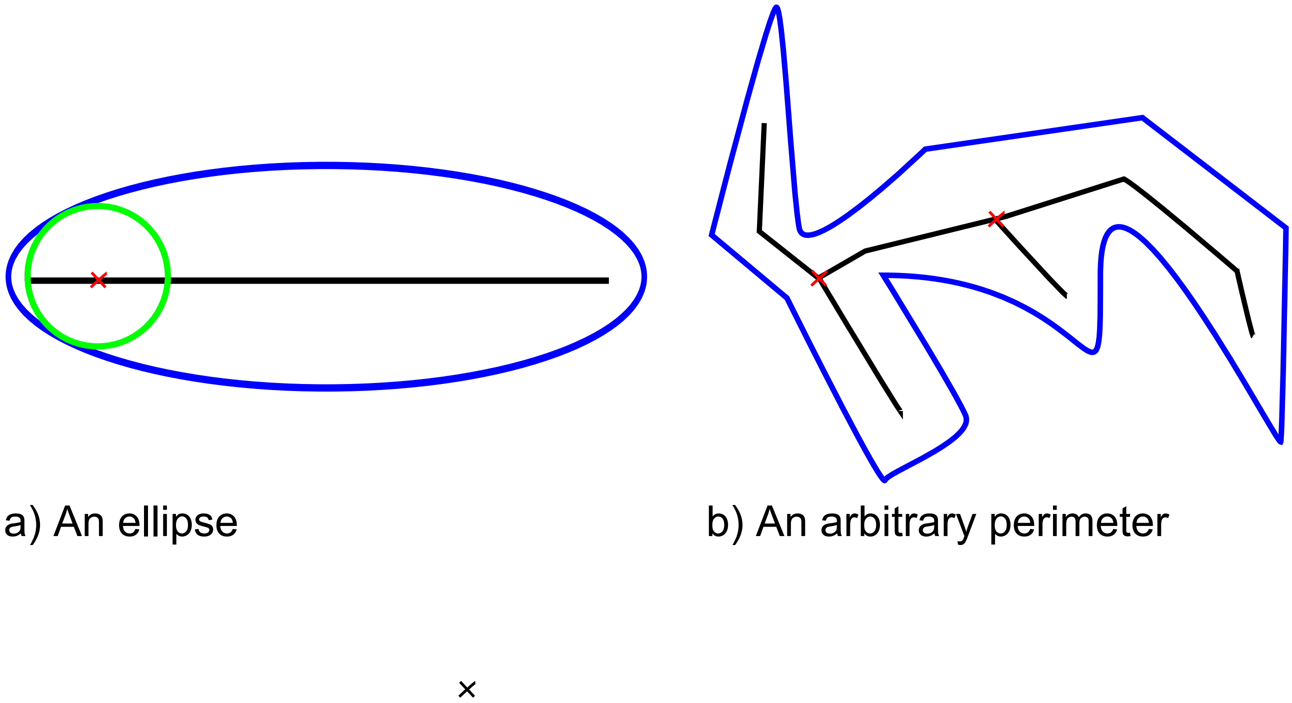

Figure 6: The medial axis skeleton (in black) for a) an ellipse and b) an arbitrary perimeter. In a) the red cross shows the centre of the green circle (which is tangent to the perimeter at two points.) In b) the red crosses show branch points in the medial axis skeleton.