Chapter 22

Modeling Notch Signaling: A Practical Tutorial

Pau Formosa-Jordan and David Sprinzak

Abstract

Theoretical and computational approaches for understanding different aspects of Notch signaling and Notch dependent patterning are gaining popularity in recent years. These in silico methodologies can provide dynamic insights that are often not intuitive and may help guide experiments aimed at elucidating these processes. This chapter is an introductory tutorial intended to allow someone with basic mathematical and computational knowledge to explore new mathematical models of Notch-mediated processes and perform numerical simulations of these models. In particular, we explain how to define and simulate models of lateral inhibition patterning processes. We provide a Matlab code for simulating various lateral inhibition models in a simple and intuitive manner, and show how to present the results from the computational models. This code can be used as a starting point for exploring more specific models that include additional aspects of the Notch pathway and its regulation.

Key words Mathematical modeling, Simulations, Lateral inhibition, Notch signaling, *Cis*-interactions, Cell-to-cell communication, Pattern formation

1 Introduction

The Notch signaling pathway has been shown to exhibit a great variety of complex behaviors in different developmental contexts [1, 2]. For instance, Notch signaling drives mutual inhibitory feedback between cells, known as lateral inhibition. This behavior leads to prototypical salt-and-pepper differentiation patterns with alternating fates in different animal tissues [3, 4]. Lateral inhibition involves competition between neighboring cells, where one cell within a group of initially equivalent cells "wins" the competition, differentiates first, and inhibits all its neighbors from differentiating themselves. The Notch-mediated inhibitory signal between the neighbors can be described by the following simplified regulatory feedback loop: Delta ligand in one cell binds to the Notch receptor on the membrane of a neighboring cell, a process that has been termed *trans*-interaction. Then, a proteolytic cleavage occurs, which releases the Notch intracellular domain (NICD) in the cell harboring the receptor. NICD serves as a co-transcription factor

Hugo J. Bellen and Shinya Yamamoto (eds.), *Notch Signaling: Methods and Protocols*, Methods in Molecular Biology, vol. 1187, DOI 10.1007/978-1-4939-1139-4_22, © Springer Science+Business Media New York 2014

that activates repressors of the Delta ligand which in turn can activate Notch signaling and its downstream targets in the neighboring cell. Different systems may vary from one to the other by having different regulatory circuits (e.g., through different target genes, or through different ligands) or tissue morphologies, what can lead to different spatiotemporal organizations in a tissue.

What are the implications of different regulatory network architectures on Notch-mediated patterning? Interpreting biological experiments is sometimes very difficult, since genetic regulatory networks can become very complex and involve counterintuitive feedback mechanisms. In the past few years, different modeling approaches have provided novel insights on how different elements in the Notch regulatory network might be operating and on the implications of these architectures in patterning (see for instance [5–24]). These in silico approaches can often provide a complementary understanding of the experimental studies, enabling the formulation of new predictions that can be experimentally tested.

This chapter is an introductory tutorial on how to start modeling some of the characteristic circuitry elements of Notch-mediated patterning. We will focus on modeling the basic elements driving lateral inhibition. This tutorial is intended for readers coming from a more biological background, with some basic mathematical and computational knowledge, that are willing to get introduced into the world of modeling Notch signaling in a practical way. Some examples of Matlab code are provided so that the reader can use it as a starting point for exploring Notch-mediated patterning. We strongly recommend though not to "copy and paste" the code from here to matlab, but to download it directly from https://github.com/dsprinzak. After reading this chapter, one should be able to model some of the basic components of Notch signaling in different kinds of cell lattices, perform numerical simulations in Matlab, and visualize the results.

The structure of the chapter is as follows. First, in Subheading 1, we present the basic mathematical model developed for studying lateral inhibition in two cells. We then generalize it to lateral inhibition in regular cell lattices. Afterwards, in Subheading 2, we introduce a more realistic model in which proteolytic cleavage of receptors and ligands occurs and take into account interactions between receptors and ligands within the same cell, what is know as *cis*-interactions. We will also show an example in which cell-to-cell interactions are mediated by longer range interactions (e.g., through filopodia). Additionally, we will provide an example where Notch signaling is modulated by an external morphogen gradient in the tissue. In Subheading 3 we will briefly discuss different sources of cell-to-cell variability that are being implemented in recent models of Notch signaling, comment on modeling additional Notch intracellular regulatory elements, and provide additional references for further reading.

Delete "will"

to add "finally in Subheading 4 we"

2 Methods

2.1 A Phenomenological Approach to Lateral Inhibition: The Collier Model

2.1.1 Lateral Inhibition in Two Cells

One of the first theoretical models for lateral inhibition dynamics was proposed by Collier and coworkers in 1996 [5]. This is a simplified model of two ordinary differential equations per cell. In each cell (cell i) one equation models the dynamics of Delta concentration, D_i, and the other equation accounts for the dynamics of repressor concentration, R_i. The Collier model basically assumes that lateral inhibition feedback is mediated by two regulatory processes: (1) Delta in each cell activates, through Notch signaling, the repressor in the neighboring cell, and (2) the repressor in each cell downregulates Delta expression in the same cell (Fig. 1a). This model uses Hill-type functions [25] to describe the activation and repression. Hill functions are monotonically increasing or decreasing sigmoidal functions which are widely used for modeling regulatory networks [25]. Apart of the Hill-type functions, each equation contains a normal linear degradation term, accounting for the typical half-life of every species (D_i and R_i). The lateral inhibition circuit for the two-cell system is therefore given by

$$\frac{\mathrm{d}D_1}{\mathrm{d}\tau} = \frac{\alpha_\mathrm{d}}{1+\left(\frac{R_1}{\theta_\mathrm{r}}\right)^h} - \gamma_\mathrm{d} D_1 \qquad (1)$$

$$\frac{\mathrm{d}R_1}{\mathrm{d}\tau} = \frac{\alpha_\mathrm{r} D_2^m}{\theta_\mathrm{d}^m + D_2^m} - \gamma_\mathrm{r} R_1 \qquad (2)$$

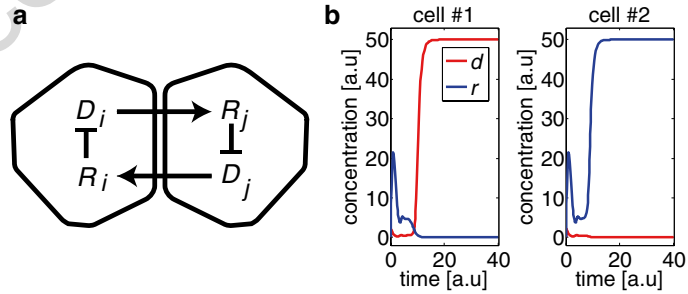

Fig. 1 Lateral inhibition between two cells: the Collier model. (**a**) A scheme showing the lateral inhibition feedback loop in a model of two species per cell; the Delta ligand levels, *D*, and the repressor levels, *R*. *Normal arrows* denote activation, *blunt arrows* denote inhibition. A cell expressing Delta ligand activates the production of repressor in its neighboring cell, which represses the production of further ligand in such cell. (**b**) Simulation results for the Collier model in a two cell system (Eqs. 5–6). [a. u.] denotes arbitrary units. Further simulation details can be found in the text. Parameter values are written in the corresponding *param* function

for cell #1 and

$$\frac{\mathrm{d}D_2}{\mathrm{d}\tau} = \frac{\alpha_\mathrm{d}}{1+\left(\frac{R_2}{\theta_\mathrm{r}}\right)^h} - \gamma_\mathrm{d} D_2 \tag{3}$$

$$\frac{\mathrm{d}R_2}{\mathrm{d}\tau} = \frac{\alpha_\mathrm{r} D_1^m}{\theta_\mathrm{d}^m + D_1^m} - \gamma_\mathrm{r} R_2 \tag{4}$$

for cell #2, where τ is time, γ_x and α_x are the degradation and maximal production rates for the x-species, θ_r is the threshold of repressor for inhibiting the ligand production to its half-value and θ_d is the threshold of Delta concentration for inducing half production of repressor in the neighboring cell, and m and h are the exponents for the activatory and inhibitory functions, respectively. In order to reduce the number of parameters, it is worth to nondimensionalize the system of Eqs. 1–4. We perform the change of variables by doing $\tau = T_0 t$, $D_i = D_0 d_i$, and $R_i = R_0 r_i$, where T_0, D_0, and R_0 are characteristic dimensional quantities of time, ligand, and repressor concentration, and t, d_i, and r_i are the nondimensional time, ligand, and repressor concentration, respectively. The nondimensionalization (i.e., the particular choice of T_0, D_0, and R_0) can be performed in different ways [26], and the modeler has to chose the one that is more convenient in relation to the questions to be answered. By choosing $T_0 = 1/\gamma_\mathrm{r}$, $D_0 = \theta_\mathrm{d}$ and $R_0 = \theta_\mathrm{r}$ we obtain the following nondimensionalized system of equations for cell i,

$$\frac{\mathrm{d}d_i}{\mathrm{d}t} = \nu \left\{ \frac{\beta_\mathrm{d}}{1+r_i^h} - d_i \right\} \tag{5}$$

$$\frac{\mathrm{d}r_i}{\mathrm{d}t} = \frac{\beta_\mathrm{r} d_j^m}{1+d_j^m} - r_i, \tag{6}$$

with $i,j = 1,2$ $i \neq j$, and where $\beta_\mathrm{d} = \alpha_\mathrm{d}/\gamma_\mathrm{d}\theta_\mathrm{d}$ and $\beta_\mathrm{r} = \alpha_\mathrm{r}/\gamma_\mathrm{r}\theta_\mathrm{r}$, so β_d is related to the ligand production and β_r to the strength of *trans-activation* due to cell-to-cell interactions. ν is a ratio of the ligand and repressor degradation rates, i.e., $\nu = \gamma_\mathrm{d}/\gamma_\mathrm{r}$, or equivalently, the typical timescale of repressor dynamics with respect to the timescale of ligand dynamics. Note that different nondimensionalizations have been used in other studies (see for instance [5, 23]). After the nondimensionalization, we have just five parameters. We can easily investigate the behavior of two of them, β_d and β_r, and relate it with experimental perturbations where Delta expression is varied, and in which the processing rate of *trans*-interactions is disrupted, for instance, through Notch inhibitor treatment [17].

In the simulation, we want to numerically solve these four equations. For doing that, we use a code written in Matlab, made of different functions (see the code below). There is a main function, in this case *twocell_LI* function, which calls other functions to perform the simulations. This function has the following structure:

1. Define the parameters of the system. Parameters are set through *params* structure.
2. Call the connectivity matrix M that indicates which cells are neighbors. This is a $k \times k$ symmetric matrix, where k is the number of cells. Position ij in the matrix (i.e., in the ith column and jth row) gets a value of 1 if cell i is a neighbor of cell j, and 0 otherwise. In the case of two cells ($k=2$), the connectivity matrix reads

$$M = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{7}$$

   In this case this matrix describes a simple situation where cell #1 is the neighbor of cell #2, and vice versa. In the code below the connectivity matrix is defined in *getconnectivityM* function. Note that the Delta levels in the neighboring cell(s) to cell i, denoted by $\langle d_i \rangle$, can be represented by the following algebraic equation:

$$\begin{pmatrix} \langle d_1 \rangle \\ \langle d_2 \rangle \end{pmatrix} = M \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \tag{8}$$

   This notation simplifies the code in the next sections.
3. Set the initial conditions. Here we choose initial repressor levels to be zero, while initial Delta levels are set to low values with some small variability or noise;

$$d_i(t=0) = \epsilon \beta_d (1 + \sigma U_i), \tag{9}$$

   where ϵ being a small number ($\epsilon = 10^{-5}$), σ is the noise amplitude ($\sigma = 0.2$) and U_i being a uniform random number between -0.5 and 0.5. This is set through *getIC* function.
4. Numerically solve the differential equations by using a standard numerical equation solver of Matlab, *ode23*. Function *li* in the code contains the differential equations for Delta and repressor concentration levels for each of the cells. This solver gets as an argument the *li* function, the time span for simulation, the initial conditions, and the parameters.
5. Plot the results. Function *plot2cells* plots Delta and repressor levels as a function of time.

Table 1
Model variables and parameters used in the text (first column), and its correspondence in the Matlab code (second column) if any, followed by a brief explanation (third column)

Model nomenclature	Code nomenclature	Definitions and comments
β_d, β_n, and β_r	betaD, betaN, and betaR	Delta, Notch, and repressor nondimensional productions
ν and μ	nu and mu	Delta and Notch degradation ratios with respect to repressor degradation
h and m	m and h	Cooperativity for Delta inhibition and repressor activation
k_t and k_c	kt and kc	*Trans*-annihilation and *cis*-inactivation nondimensional strengths
d_i, r_i, and n_i	D, R, and N	Levels of Delta, repressor, and Notch receptor concentration for the i cell (model) and for all cells (code)
$\langle d_i \rangle$ and $\langle n_i \rangle$	Dneighbor and Nneighbor	Average of Delta and Notch receptor concentration for the i cell (model) and for all cells (code)
$d_i(t=0)$, $r_i(t=0)$, and $n_i(t=0)$	D0, R0, and N0	Initial conditions for Delta, repressor, and Notch levels for the i cell (model) and for all cells (code)
ϵ and σ	Epsilon and sigma	Parameters related to the noise in the initial conditions
P, Q, k, w, and M	P, Q, k, w, and M	Cell lattice parameters and connectivity matrix
	Tmax	Maximum time for a simulation
	l	Length scale of the gradient

The code is the following (see also Table 1):

```
function [yout,tout,params] = twocell_LI(params)
% Twocell_LI simulates lateral inhibition between
  two cells. The
% structure params contains the model parameters
  of the system.
% TOUT is a vector containing the time points of
  the solution
% between 0 and Tmax. YOUT is a matrix containing
  the numerical
% solution for each variable for each time point.
  Each row in
% YOUT is a vector of the size of TOUT.
   Tmax=40;  tspan=[0  Tmax];  %  set  time  for
simulation
   k=2; % number of cells
% get the default parameters if none provided
if(nargin < 1)
    params=defaultparams;
end
```

```
% get the connectivity matrix
params.connectivity=getconnectivityM;

% setting the initial conditions (IC) + noise
y0=getIC(params,k);

% run simulation with lateral inhibition
[tout,yout] = ode23(@li,tspan,y0,[],params);

% show time traces of two cells with lateral inhibition
plot2cells(tout,yout,k)

function dy = li(t,y,params)

nu=params.nu;
betaD=params.betaD;
betaR=params.betaR;
h=params.h;
m=params.m;
M=params.connectivity;
k=length(M);

D = y(1:k); % levels of Delta in cells 1 to k
R = y(k+1:2*k); % levels of repressor in cells 1
   to k
Dneighbor=M*y(1:k);% Delta level in the neighbor-
   ing cells

% differential equations for Delta and repressor
   levels
dD = nu * (betaD.*1./(1 + R.^h)-D);
dR = betaR.*Dneighbor.^m./(1 + Dneighbor.^m)-R;

dy = [dD;dR];

function params=defaultparams

params.nu=1;        % ratio of degradation rates
params.betaD=50;      % non-dimensional Delta
production
params.betaR=50;    % non-dimensional repressor
production
params.h=3;          % Hill coefficient repression
function
params.m=3;          % Hill coefficient activating
function
params.sigma=0.2;    % noise amplitude in initial
conditions

function M=getconnectivityM

M=[0 1;1 0]; % 2 cell connectivity matrix

function y0=getIC(params,k)

U=rand(k,1)   -   1/2;   %   a   uniform   random
distribution
epsilon=1e-5;    % multiplicative factor of Delta
initial condition
```

```
D0=epsilon*params.betaD.*(1 + params.sigma*U); %
initial Delta levels
R0=zeros(k,1);  % initial repressor levels
y0=[D0;R0];     % vector of initial conditions
function plot2cells(tout,yout,k)
figure(21); clf
for i=1:2
    subplot(1,2,i)
    plot(tout,yout(:,i),'-r','linewidth',2)  % plot
    Delta levels
    hold on
    plot(tout,yout(:,k+i),'-b','linewidth',2) % plot
    repressor levels
    title(['cell #',num2str(i)])
    xlabel('time [a.u]');
    ylabel('concentration [a.u]')
    legend('d','r')
end
```

This code can be expanded to larger systems and other dynamics (see code examples in the next sections).

Running the code results in Fig. 1b. Both cells start expressing Delta and the repressor, and pass transiently through a homogeneous state, i.e., a state in which both cells have the same levels in each of its variables. This transient homogeneous state matches with the homogeneous steady state of the dynamics, i.e., the solution of $\mathrm{d}d_i/\mathrm{d}t = \mathrm{d}r_i/\mathrm{d}t = 0$ with $\langle d_i \rangle = d_i$ for every i-cell. The homogeneous state can be either stable or unstable. In the represented case, it is unstable, and the two cell system becomes patterned when Delta concentration in one cell goes up, inhibiting Delta concentration of its neighbor. In different parameter ranges, the system could stay in the unpatterned homogeneous steady state, for instance, if there is no cooperativity in the Hill functions, namely $h = 1$ and $m = 1$ [17].

2.1.2 Lateral Inhibition in a Regular Cell Lattice

Lateral inhibition often occurs over extended regions of a tissue containing many cells. It is therefore interesting to model lateral inhibition on regular cell lattices. In this case, the repressor in each cell is activated by the average ligand concentration of its neighboring cells, so now the repressor dynamics reads

$$\frac{\mathrm{d}r_i}{\mathrm{d}t} = \frac{\beta_\mathrm{r} \langle d_i \rangle^m}{1 + \langle d_i \rangle^m} - r_i, \tag{10}$$

where $\langle d_i \rangle$ has the following expression:

$$\langle d_i \rangle = \frac{1}{w} \sum_{j \in \mathrm{nn}(i)} d_j. \tag{11}$$

Here $j \in nn(i)$ refers to all j-cells that are nearest neighbors to cell i, and w is the number of nearest neighbors to cell i. For a one dimensional line of cells, the average ligand concentration (Eq. 11) will read

$$\langle d_i \rangle = \frac{1}{2}\left(d_{i+1} + d_{i-i}\right). \tag{12}$$

For squared and hexagonal two dimensional cell lattices this averaged term takes the form

$$\langle d_i \rangle = \frac{1}{4}\left(d_{(i1)} + d_{(i2)} + d_{(i3)} + d_{(i4)}\right) \tag{13}$$

and

$$\langle d_i \rangle = \frac{1}{6}\left(d_{(i1)} + d_{(i2)} + d_{(i3)} + d_{(i4)} + d_{(i5)} + d_{(i6)}\right), \tag{14}$$

where (ij) in Eqs. 13 and 14 represents the index of the jth neighbor of cell i.

Herein, we choose the hexagonal cell lattice since this is the most similar to the natural cell packing. In the *multicell_LI* code we are simulating multicellular lateral inhibition in a hexagonal cell lattice of P rows and Q columns. The code is very similar to *twocell_LI* except that now the connectivity matrix accounts for the six neighbors of each cell. In this case it is necessary to define an indexing scheme that easily allows tracking all the cells and their neighbors. Here, we switch between two indexing schemes—one that numbers the cells from 1 to k (i, Fig. 2a), and one that keeps the row and column of each cell (p, q, Fig. 2b). Each element of the connectivity matrix is multiplied by $1/w$, with w being the number of nearest neighbors (e.g., $w = 6$).

In order to avoid boundary effects in the simulations (e.g., cells at the edge may behave differently than cells in the middle), we normally use periodic boundary conditions. For example, in a line of cells, we define that the two cells at the two ends of the line become nearest neighbors, so instead of a line of cells we get a ring of cells. In such a ring with P cells, every x species (Delta or repressor) satisfies $x_{i+P} = x_i$. Similarly, for a two dimensional lattice of $P \times Q$ cells, periodic boundary conditions imply $x_{p+P,q+Q} = x_{p,q}$, so the cell lattice can be represented on a torus.

The code for the multicellular system becomes (see also Table 1; copy functions from earlier code where indicated):

```
function [yout,tout,params,F] = multicell_LI(params)
% multicell_LI simulates lateral inhibition in a
  hexagonal lattice.
```

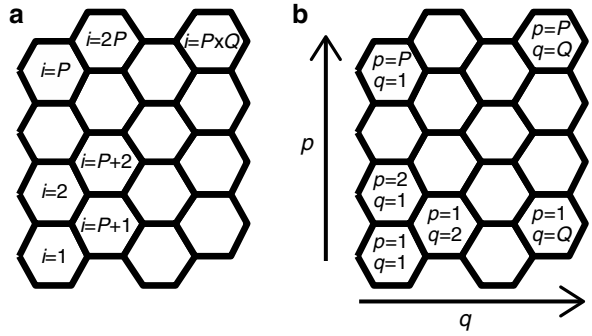

Fig. 2 Labeling schemes in a regular hexagonal cell lattice. (**a**) One index labeling scheme. (**b**) Two indices labeling scheme. Having two indices per cell facilitates the computation of the neighboring cell indices and the implementation of the periodic boundary conditions


```
% The structure params contains the model parameters
  of the system.
% TOUT is a vector containing the time points of the
  solution
% between 0 and Tmax. YOUT is a matrix containing
  the numerical
% solution for each variable for each time point.
  Each row in
% YOUT is a vector of the size of TOUT. F is a movie of the
% simulation.
Tmax=40; tspan=[0 Tmax]; % set time for simulation
% get the default parameters if none provided
if(nargin < 1)
    params=defaultparams;
end
P=params.P;      % number of cells per column
Q=params.Q;     % number of columns - MUST BE EVEN
k=P*Q;           % number of cells
% get the connectivity matrix
params.connectivity=getconnectivityM(P,Q);
% setting the initial conditions (IC) + noise
y0=getIC(params,k);
% run simulation with lateral inhibition
[tout,yout] = ode23(@li,tspan,y0,[],params);
% show time traces of two cells with lateral inhibition
plot2cells(tout,yout,k)
% show lattice simulation
F=movielattice(tout,yout,P,Q,k);
```

```
function dy = li(t,y,params)
[USE THE SAME FUNCTION AS TWOCELL_LI]

function params=defaultparams
params.nu=1;         % ratio of degradation rates
params.betaD=50;     % normalized Delta production
params.betaR=50;   % normalized repressor production
params.h=3;          % Hill coefficient repression
function
params.m=3;          % Hill coefficient activating
function
params.sigma=0.2;    % noise amplitude in initial
conditions
params.P=18;         % number of cells per column
params.Q=18;        % number of columns - MUST BE EVEN

function M=getconnectivityM(P,Q)

k=P*Q;           % number of cells
M=zeros(k,k);    % connectivity matrix
w=1/6;           % weight for interactions

% calculating the connectivity matrix
for s=1:k
    kneighbor=findneighborhex(s,P,Q);
    for r=1:6
        M(s,kneighbor(r))=w;
    end
end

function y0=getIC(params,k)
[USE THE SAME FUNCTION AS TWOCELL_LI]

function plot2cells(tout,yout,k)
[USE THE SAME FUNCTION AS TWOCELL_LI]

function out = findneighborhex(ind,P,Q)

% This function finds the 6 neighbors of cell ind
[p,q] = ind2pq(ind,P);

% above and below:
out(1) = pq2ind(mod(p,P)+1,q,P);
out(2) = pq2ind(mod(p-2,P)+1,q,P);

% left and right sides:
qleft = mod(q-2,Q)+1;

qright = mod(q,Q)+1;

if q/2~=round(q/2),
    pup = p;
    pdown = mod(p-2,P)+1;
else
    pup = mod(p,P)+1;
    pdown = p;
end;
```

```
out(3) = pq2ind(pup,qleft,P);
out(4) = pq2ind(pdown,qleft,P);
out(5) = pq2ind(pup,qright,P);
out(6) = pq2ind(pdown,qright,P);

function ind=pq2ind(p,q, P)
ind = p + (q-1)*P;

function [p,q]=ind2pq(ind, P)
q = 1+floor((ind-1)/P);
p = ind - (q-1)*P;

function plotHexagon(p0,q0,c)

% This function plots a hexagon centered at coor-
dinates p,q
s32 = sqrt(3)/4;
q = q0*3/4;
p = p0*2*s32;
if q0/2 == round(q0/2),
   p = p+s32;
end;

x(1)=q-.5; x(2)=q-.25; x(3)=q+.25;
x(4)=q+.5; x(5)=q+.25; x(6)=q-.25;

y(1)=p ; y(2)=p+s32; y(3)=p+s32;
y(4)=p; y(5)=p-s32; y(6)=p-s32;

patch(x,y,c,'linewidth',2);

function F=movielattice(tout,yout,P,Q,k)

% This function generates a movie of patterning in
  a hexagonal
% lattice. The color represents the level of Delta.
  It also
% saves the movie as an AVI file.

figure(22)
Cmax=max(yout(end,1:k)); % finds max(Delta) at the
end point
frameind=0;
for tind = 1:5:length(tout),   % shows every 5th frame
    clf;
    for i = 1:P,
        for j = 1:Q,
          ind = pq2ind(i,j,P);
          mycolor = min([yout(tind,ind)/Cmax,1]);
           plotHexagon(i,j,[1-mycolor,1-mycolor,1]);
        end;
    end;
    axis image; axis off; box off;
    frameind=frameind+1;
    F(frameind) = getframe; % generates a movie
    variable
end;
```

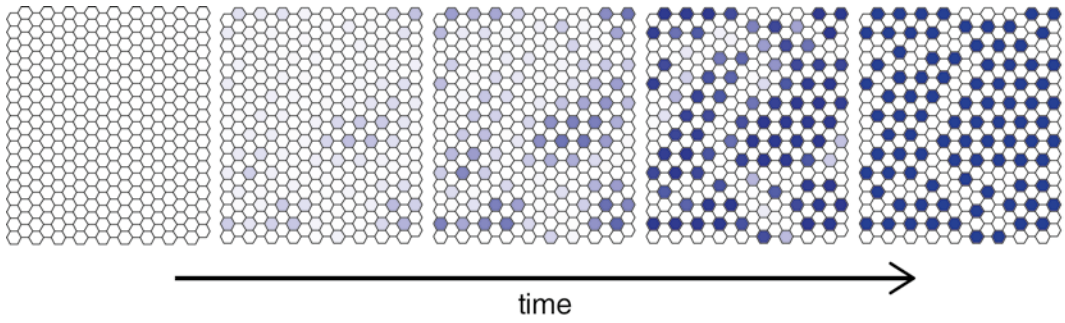

Fig. 3 Snapshots at different time points of a simulation for the Collier model (Eqs. 5, 10 and 14) in a hexagonal cell lattice with periodic boundary conditions. *Blue* intensity denotes the Delta levels. *Dark blue* corresponds to $d_i = 50$, while *white* corresponds to $d_i = 0$. From *left* to *right*, the time points shown are $t = 0$, $t = 16.0$, $t = 16.7$, $t = 18.1$, and $t = 29.6$ in arbitrary units. Further simulation details can be found in the text. Parameter values can be found in the corresponding *param* function

Could we place this figure at the bottom of this page instead? (so it appears after it is cited)

```
% save movie in avi format
movie2avi(F,'movielattice','compression','none');
```

In Fig. 3 we can see how patterning spontaneously emerges in a hexagonal cell lattice from an initially uniform state. Note that the final simulation state exhibits domains of ordered patterns separated by gaps or defects. This is typically the case for Collier type models [5].

Often, apart from performing simulations with a single set of parameters, one is interested in performing extensive simulations across a two dimensional parameter space, i.e., to perform simulations by varying two parameters while maintaining the rest fixed. In these explorations it is useful to define an observable that allows characterization of the resulting phenotype. A possible observable would be the density of high ligand cells in the cell lattice [23], which provides an idea of the ratio between number of cells from each type, or the logarithm of the ratio between high Delta cells and low Delta cells, to distinguish patterned regions from homogeneous regions in the parameter space. Another interesting observable is a measure of the time required for patterning, which can reveal how the dynamics of the system is affected by the different parameters.

The following code calls *multcell_LI* function in a new β_d and β_r parameter set each time and plots a phase diagram of the last two aforementioned observables in the β_d and β_r parameter space (Fig. 4). It is suggested to comment out the plotting functions in *multicell_LI* to shorten the running time of the code and to change *Tmax* to a larger value (to also capture slower dynamics).

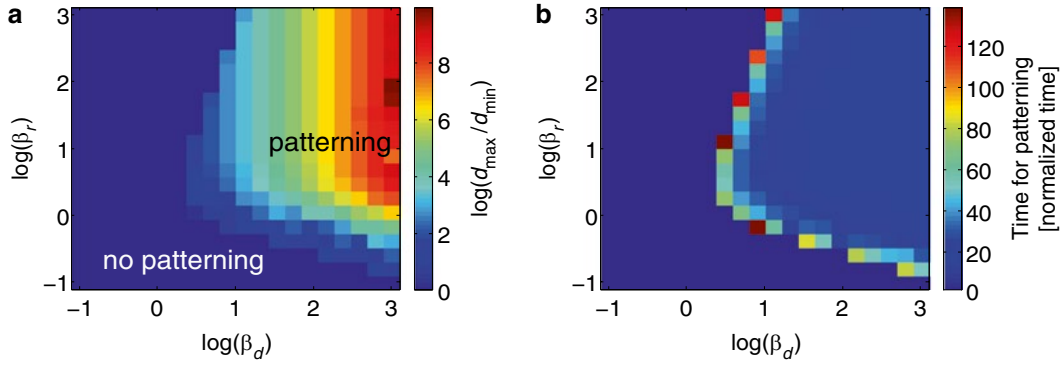

Fig. 4 Parameter space analysis for the Collier model (Eqs. 5, 10, and 14). (**a**) A phase diagram showing $\log(d_{max}/d_{min})$ for different values of the parameters β_r and β_d, where d_{max} and d_{min} are the maximal and minimal nondimensional Delta levels in the steady state. The color bar indicates the color code corresponding to the value of $\log(d_{max}/d_{min})$. This figure shows the region in parameter space where the homogeneous solution is found (*dark blue region* with the label "no patterning") and the region where patterning emerges (corresponding to the remaining colored region with the label "patterning"). (**b**) A phase diagram showing the time required for patterning for each parameter set where patterning occurs. The color bar indicates the color code corresponding to patterning time values (see text for definition of patterning time). Simulations were performed with Tmax = 200. This figure was generated by running *paramsearch_LI* code given in the text

Missing full stop "."

putting "multicell_LI function" into the precedent line so that it reads "MOVIELATTICE within multicell_LI function" in a single sentence

```
464  function paramsearch_LI
465  % This code plots the log(Dmax/Dmin) and the time
466    required for patterning for different betaD and
467    betaR.
468  % FOR A FASTER RUN, COMMENT OUT PLOT2CELLS AND
469    MOVIELATTICE within
470    multicell_LI function
     % fixed parameters
     params.nu=1;
     params.h=3;
     params.m=3;
     params.sigma=0.2;
     params.Q=12;
     params.P=12;
     k=params.Q*params.P;
479  % variable parameters
480  betaD=logspace(-1,3,20); % creates a series of betaD
481  from 0.1 to 1000
482  betaR=logspace(-1,3,20); % creates a series of betaR
483  from 0.1 to 1000
484  ind=0; h=waitbar(0,'% of progress'); % generates a
485  waitbar
486  for i=1:length(betaD)
487      for j=1:length(betaR)
488          params.betaD=betaD(i);
489          params.betaR=betaR(j);
490          ind=ind+1;
491          waitbar(ind/(length(betaD)*length(betaR)))
```

```
            [yout,tout] = multicell_LI(params); % calling the LI solver
            % finding max and min values of D
            Dmax(i,j)=max(max(yout(end,1:k)));
            Dmin(i,j)=abs(min(min(yout(end,1:k))));
            % finding cases where patterning occurs (when Dmax/Dmin>1.2)
            % and getting the patterning time
            if Dmax(i,j)/Dmin(i,j)>1.2
                T(i,j)=getPatterningTime(tout,yout,...
                k,Dmax(i,j),Dmin(i,j));
            else
                T(i,j)=NaN; % patterning time is not set for the no patterning case
            end
        end
    end
    close(h)
    figure(23)
    imagesc(log10(betaD),log10(betaR),...log10(Dmax./Dmin));
    set(gca,'YDir','normal')
    xlabel('log(\beta_d)','fontsize',14);
    ylabel('log(\beta_r)','fontsize',14);
    title('log(d_{max}/d_{min})','fontsize',14)
    colorbar
    figure(24)
    imagesc(log10(betaD),log10(betaR),T);
    set(gca,'YDir','normal')
    xlabel('log(\beta_d)','fontsize',14);
    ylabel('log(\beta_r)','fontsize',14);
    figtit='Time for patterning [normalized time]'title(figtit,'fontsize',14)
    colorbar
    function T=getPatterningTime(tout,yout,k,Dmax,Dmin)
    % This function estimates the time required for patterning.
    % This is done by the following 3 steps:
    % 1. find all the high D cells ('onCells')
    % 2. find the time it takes for each 'on cell' to reach 90% of its final level ('TonCells')
    % 3. get median value of the times calculated in stage 2
    onCells=find(yout(end,1:k)>0.5*(Dmax+Dmin));
    for i=1:length(onCells)
```

Please write
"log10(Dmax./"
in the following line, so
line 512 reads
"log10(Dmax./Dmin));"

See note below (*)

(*) Please, instead of lines 523-524, write the following in 4 lines, as it is:

```
figtit0='Time for patterning '
figtit1='[normalized time]'
figtit=[figtit0,figtit1]
title(figtit,'fontsize',14)
```

```
    Tind=find(yout(:,onCells(i))>0.9*yout(end,onCe
    lls(i)),1,'first');
    TonCells(i)=tout(Tind);
end
T=median(TonCells);
```

In Fig. 4 we can see that patterning occurs in a wide range of β_d and β_r values, which means that it is a very robust process. Figure 4b shows how patterning time varies with β_d and β_r values. The patterning time is significantly increased at the edge of the patterning region, a behavior known as critical slowing down [27]. The code can be easily adapted to plot the dependence on any two parameters of the model and to be used in any of the models provided in this tutorial.

2.2 Extensions to the Collier Model

2.2.1 Adding Trans-Annihilation and Cis-Inactivation

So far we have simulated a simplified model, which does not take into account some of the biochemistry of Notch signaling. More kinetic-based models can also be used, which takes into account the cleavage of Notch, the endocytosis of Delta, and the *cis-interaction* between Notch and Delta [28]. The latter interaction has been shown to lead to mutual inactivation of both Notch and Delta [17, 18, 29]. To account for these processes we need to add to our model the level of Notch receptor concentration in a cell, given by the variable N_i. These processes modify the lateral inhibition model, which leads to the following differential equations:

$$\frac{dN_i}{d\tau} = \alpha_n - K_t N_i \langle D_i \rangle - K_c N_i D_i - \gamma_N N_i \tag{15}$$

$$\frac{dD_i}{d\tau} = \frac{\alpha_d}{1+\left(\frac{R_i}{\theta_r}\right)^b} - K_t D_i \langle N_i \rangle - K_c N_i D_i - \gamma_D D_i \tag{16}$$

$$\frac{dR_i}{d\tau} = \frac{\alpha_r \left(\frac{K_t N_i \langle D_i \rangle}{\gamma_{nd}}\right)^m}{\theta_{nd}^m + \left(\frac{K_t N_i \langle D_i \rangle}{\gamma_{nd}}\right)^m} - \gamma_R R_i, \tag{17}$$

where $\langle N_i \rangle$ and $\langle D_i \rangle$ are the average receptor and ligand concentrations in the neighboring cells (see Eq. 11), so terms $K_t\ N_i \langle D_i \rangle$ and $K_t\ D_i\ \langle N_i \rangle$ denote *trans*-annihilation (cleavage of Notch and endocytosis of Delta), while $K_c\ N_i\ D_i$ denote *cis*-inactivation. The strength of these interactions is parameterized by K_t and K_c, respectively. γ_{nd}^{-1} is a typical timescale of the *trans*-complex, and θ_{nd} represents a typical amount of *trans*-complex for activating the

repressor. More details about the derivation of these equations can be found in [17, 18].

We perform the nondimensionalization of Eqs. 15–17 by following the same steps as we did in a precedent section, with now $N_i = N_0 n_i$, where N_0 is a characteristic dimensional quantity of receptor concentration, and n_i is the nondimensional receptor concentration in the i cell. Now we set $T_0 = 1/\gamma_r$, $N_0 = \theta_{nd}\,\gamma_{nd}/\gamma_n$, $D_0 = \theta_{nd}\,\gamma_{nd}/\gamma_d$, and $R_0 = \theta_r$, so the resulting nondimensional system reads

$$\frac{dn_i}{dt} = \mu\left\{\beta_n - k_t n_i \langle d_i \rangle - k_c n_i d_i - n_i\right\} \tag{18}$$

$$\frac{dd_i}{dt} = \nu\left\{\frac{\beta_d}{1+r_i^b} - k_t d_i \langle n_i \rangle - k_c n_i d_i - d_i\right\} \tag{19}$$

$$\frac{dr_i}{dt} = \frac{\beta_r \left(k_t n_i \langle d_i \rangle\right)^m}{1+\left(k_t n_i \langle d_i \rangle\right)^m} - r_i, \tag{20}$$

where $\mu = \gamma_n/\gamma_r$, $\nu = \gamma_d/\gamma_r$, $k_t = K_t\gamma_{nd}\theta_{nd}/(\gamma_d\gamma_n)$, $k_c = K_c\gamma_{nd}\theta_{nd}/(\gamma_d\gamma_n)$, $\beta_r = \alpha_r/\gamma_r\theta_r$, $\beta_d = \alpha_d/\gamma_{nd}\theta_{nd}$, and $\beta_n = \alpha_n/\gamma_{nd}\theta_{nd}$. Therefore, parameters μ and ν account for the timescale of receptor and ligand with respect to the repressor, respectively, k_t and k_c are the effective nondimensional strengths for *cis* and *trans*-interactions, and β_n, β_d, and β_r are effective productions of receptor, ligand, and repressor. The code implementing Eqs. 18–20 for two cells is as follows (see also Table 1; copy functions from earlier code where indicated):

"transcis2cell_LI(params)" should appear together in line 590, instead of being split between lines 589 and 590

```
function [yout,tout,params] = ...transcis2cell_LI(params)
% transcis2cell_LI simulates trans-annihilation with cis-inactivation
% between two cells. The structure params contains the model
% parameters of the system.
% TOUT is a vector containing the time points of the solution
% between 0 and Tmax. YOUT is a matrix containing the numerical
% solution for each variable for each time point. Each row in
% YOUT is a vector of the size of TOUT.
Tmax=100; tspan=[0 Tmax]; % set time for simulation
k=2; % number of cells
% get the default parameters if none provided
if(nargin < 1)
```

```
608                 params=defaultparams;
609             end
610             % get the connectivity matrix
611             params.connectivity=getconnectivityM;
612             % setting the initial conditions + noise
613             y0=getIC(params,k);
614             % run simulation with lateral inhibition
615             [tout,yout] = ode23(@li,tspan,y0,[],params);
616             % show time traces of two cells with lateral inhibition
617             plot2cells(tout,yout,k)
618             function dy = li(t,y,params)
619             nu=params.nu;
620             betaD=params.betaD;
621             betaN=params.betaN;
622             betaR=params.betaR;
623             m=params.m;
624             h=params.h;
625             M=params.connectivity;
626             k=length(M);
627             mu=params.mu;
628             kc=params.kc;
629             kt=params.kt;
630             D = y(1:k);          % levels of Delta in cells 1 to k
631             R = y(k+1:2*k);      % levels of repressor in cells 1 to k
632             N = y(2*k+1:3*k);    % levels of repressor in cells
633             1 to k
634             Dneighbor=M*y(1:k);          % Delta level in the
635             neighboring cells
636             Nneighbor=M*y(2*k+1:3*k);  % Notch level in the
637             neighboring cells
638             % differential equations for Delta, repressor, and
639             Notch levels
640             dN = mu * (betaN - kt.*N.*Dneighbor-kc.*N.*D-N);
641             dD = nu * (betaD.*1./(1 + R.^h)-kt.*D.*...Nneighbor-
642                 kc.*N.*D-D);
643             dR = betaR.*(kt.*N.*Dneighbor).^m./(1 + (kt.*N.*...
644             Dneighbor).^m)-R;
645             dy = [dD;dR;dN];
646             function params=defaultparams
647             params.nu=1;
648             params.betaD=50;
649             params.betaN=1;
650             params.betaR=200;
651             params.m=1;
```

See below (*)

(*) "Nneighbor-kc.*N.*D-D);" should appear altogether in line 641, and NOT split between lines 640 and 641

```
params.h=1;
params.sigma=0.2;
params.mu=1;
params.kc=10;
params.kt=1;

function M=getconnectivityM

M=[0 1;1 0]; % 2 cell connectivity matrix

function y0=getIC(params,k)

U=rand(k,1) - 1/2; % a uniform random
distribution
epsilon=1e-5; % multiplicative factor of Delta
initial condition
D0=epsilon*params.betaD.*(1 + params.sigma*U); %
initial Delta levels
R0=zeros(k,1); % initial repressor levels
N0=params.betaN.*ones(k,1); % initial Notch lev-
els are betaN
y0=[D0;R0;N0]; % vector of initial conditions

function plot2cells(tout,yout,k)

[USE THE SAME FUNCTION AS TWOCELL_LI]
```

By including *cis*-inactivation we can get patterning even without cooperativity, i.e., when $h=1$ and $m=1$ (data not shown, *see* refs. 17, 18). It is also easy to demonstrate (for example by running the paramsearch_LI code) that the dynamics are strongly affected by *cis*-interactions [18]. A recent work proposing an alternative more Collier-based mathematical model of cis-interactions can be found in [30].

2.2.2 Simulations with Longer Range Interactions

Recent work has shown that filopodia and cellular protrusions can take place during lateral inhibition, giving rise to sparser patterns [16]. To include these effects in our modeling framework, we have to take into account cell-to-cell interactions that can also reach cells that are further apart in the cell lattice, for example by allowing interactions between a cell and its next nearest neighbors. In the Collier model formulation, a very simple way of taking it into account would be by extending the cell-to-cell interaction in the following way:

$$\langle d_i \rangle = \frac{1}{w}\left(\sum_{j\in \mathrm{nn}(i)} d_j + \sum_{j\in \mathrm{nnn}(i)} d_j \right), \qquad (21)$$

where now nn(i) and nnn(i) refer to nearest and next nearest neighbors to cell i, and w is the total number of nearest and next nearest neighbors ($w=18$ in a regular hexagonal lattice). A more realistic cell-to-cell coupling can be found in [16]. In this case, to compute the connectivity matrix and the indices of cells contributing

"largespacing_LI(params)" should appear together and non split in line 697

in Eq. 21 we have the following code (see also Table 1; copy functions from earlier code where indicated):

```
function [yout,tout,params,F] = ...largespacing_LI(params)
[USE THE SAME FUNCTIONS IN multicell_LI REPLACING ONLY THE getconnectivityM FUNCTION]
function M=getconnectivityM(P,Q)
k=P*Q; % number of cells
M=zeros(k,k); % connectivity matrix
w=1/18; % weight for interactions
% calculating the next nearest neighbor connectivity matrix
for s=1:k
    % find the neighbors of cell s
    kneighbor=findneighborhex(s,P,Q);
    nn_neighbor=kneighbor;
    % find the neighbors of the neighbors of cell s
    for i=1:length(kneighbor)
        nn_neighbor=[nn_neighbor;...findneighborhex(kneighbor(i),P,Q)];
    end
    % find all the unique neighbors of cell s
    nn_neighbor=unique(nn_neighbor);
    for r=1:length(nn_neighbor);
        M(s,nn_neighbor(r))=w;
    end
    M(s,s)=0; % removing cell s from the connectivity matrix
end
```

"findneighborhex(kneighbor(i),P,Q)];" should appear alone in line 713 and not split between lines 712 and 713

From running this code we can see that lateral inhibition with longer range cell-to-cell interactions drives a sparser salt-and-pepper pattern of high Delta cells (see Fig. 5a) than the pattern obtained from the Collier model (see Fig. 3).

2.2.3 Adding External Gradients

Notch-mediated patterning often involves cues from other signaling systems, which may introduce long-range spatial modulation of Notch pathway components. The Notch pathway has been shown to be modulated by morphogens like Wnt, Hegdehog, EGF, among others [31, 32]. To study the effect of long-range morphogen gradients on lateral inhibition patterning, we consider a situation where a radial exponential gradient of a certain morphogen drives Delta production on a two dimensional hexagonal lattice.[1] As a first approximation, we can omit the diffusing

[1] Here we are introducing the steady state profile of a radially diffusing morphogen in two dimensions that is linearly degraded. Note that the corresponding steady state of the morphogen would follow a modified Bessel function of second kind [33], but here we use an exponential decay for simplicity.

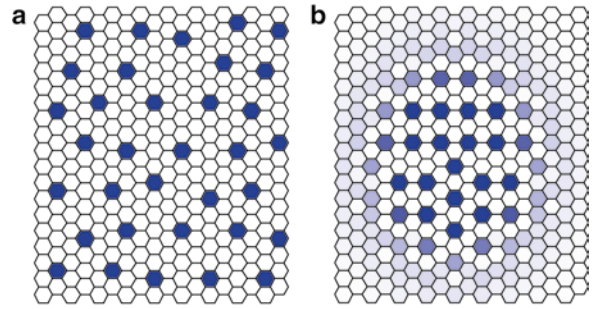

Fig. 5 Simulations with models taking into account longer range interactions and a spatial gradient. (**a**) Collier model with additional next nearest neighbors interactions (Eqs. 5, 10 and 21) drives sparser patterns of higher Delta cells. (**b**) Collier model with an exponential spatial modulation of the Delta production parameter (β_d) in the tissue. This situation emulates a scenario in which Delta is activated downstream a radial morphogen gradient that exponentially decays from the center of the tissue. This enables the creation of a localized patterning domain in the tissue. Color codes as in Fig. 3. *Dark blue color* corresponds to $d = 50$ in panel **a** while $d = 2.96$ in panel **b**. In panel **b**, the *darkest blue* intensity has been assigned to the 95th percentile of the Delta levels in the cell lattice at the steady state, so Delta levels larger than $d = 2.96$ also have been depicted by the same *dark blue color*. Further simulation details can be found in the text. Parameter values can be found in the corresponding *params* structure

morphogen and focus directly on modeling its downstream effect as a spatial modulation of the Delta production parameter in the Collier model. In the following simulation code, β_d is multiplied by an exponential function with lengthscale l, so that the production of Delta varies from cell to cell (see also Table 1; copy functions from earlier code where indicated):

```
function [yout,tout,params,F] = morphogen_LI(params)
% morphogen_LI simulates lateral inhibition in a
  hexagonal lattice.
% The morphogen is introduced through a gradient
  on betaD.
% The structure params contains the model parame-
  ters of the system.
% TOUT is a vector containing the time points of
  the solution
% between 0 and Tmax. YOUT is a matrix containing
  the numerical
% solution for each variable for each time point.
  Each row in
% YOUT is a vector of the size of TOUT. F is a
  movie of the simulation.

Tmax=30; tspan=[0 Tmax]; % set time for simulation
```

```
% get the default parameters if none provided
if(nargin < 1)
    params=defaultparams;
end

P=params.P;  % number of cells per column
Q=params.Q;  % number of columns - MUST BE EVEN
k=P*Q; % number of cells

% get the connectivity matrix
params.connectivity=getconnectivityM(P,Q);

% apply morphogen controlling betaD.
% params.l is the lengthscale set by a morphogen
params.l=1.5;
Morph=getMorph(params.l,P,Q);
% params.betaD becomes a vector describing the
local production
% of Delta controlled by a morphogen

params.betaD=params.betaD*Morph;

% setting the initial conditions + noise
y0=getIC(params,k);

% run simulation with lateral inhibition
[tout,yout] = ode23(@li,tspan,y0,[],params);

% show time traces of two cells with lateral inhibition
plot2cells(tout,yout,k)

% show lattice simulation
F=movielattice(tout,yout,P,Q,k);

[USE THE SAME FUNCTIONS AS IN multicell_LI ADDING
ONLY getMorph FUNCTION]

function Morph=getMorph(l,P,Q)

% This function generates an exponential morphogen
profile with lengthscale params.l
center=[floor(P/2) floor(Q/2)];
MorpPQ=zeros(P,Q);
Morph=zeros(P*Q,1);
for p=1:P
    for q=1:Q
        distpq=sqrt((((p-center(1))/l)...^2)+
        (((q-center(2))/l)^2));
        MorphPQ(p,q)=exp(-distpq);
        ind=pq2ind(p,q, P);
        Morph(ind)=MorphPQ(p,q);
    end
nd
```

"^2)+ (((q-center(2))/l)^2));" should appear alone in line 794 and NOT split between lines 794 and 795

In Fig. 5b we can see that such radial morphogen gradient pstream of Delta can restrict the lateral inhibition pattern to a

certain tissue domain. This could be a plausible mechanism to set the size of the domain of lateral inhibition patterns.

Other examples of parameter modulation across a cell lattice in lateral inhibition dynamics can be found in [34, 35]. More complex models that explicitly take into account a diffusing morphogen that affects the patterning process can be found in the context of differentiation wavefronts [23, 36–38].

3 Notes

3.1 Adding Cell-to-Cell Variability

Cell-to-cell variability can be manifested in different ways in a tissue during the patterning process. Herein we will just mention some examples that have already been considered in models for lateral inhibition.

Cells in a tissue can have different number of neighbors, so working with cell lattices with a certain degree of irregularity, e.g., Voronoi tessellations, could capture such heterogeneity in the number of first neighbors [9, 10, 16, 23, 39, 40]. One step further is to consider the connectivity matrix as a dynamic one [16, 41]. This has already been used for modeling the highly dynamic nature of filopodia, and it has been shown to have an effect in the refinement of the final pattern [16]. This dynamic cell-to-cell connectivity has been referred as structured noise [41].

Another source of cell-to-cell variability is cells having different contact areas among them due to heterogeneity in its shape. This can be set through an irregular cell lattice in which the strength of *trans*-interactions is proportional to each cell-to-cell contact area [9, 10, 23].

Cell-to-cell variability can also be taken into account through static heterogeneity in the model parameters [18]. Another source of variability may come from fluctuations in the levels of the molecular components of the pathway [42], e.g., receptors and ligands, and other molecular components in the cell. One can consider this effect by using stochastic differential equations in the Itô approximation [43]. This kind of dynamical noise has been implemented in different models of Notch signaling in different ways [23, 38, 44].

3.2 Modeling Additional Intracellular Regulatory Elements

Recent theoretical works have modeled downstream targets of Notch, or upstream regulators of Notch and its ligands [7, 11, 24, 45–48]. These elements have been modeled as separate small modules, and also have been embedded in larger models of Notch signaling. Note that adding more variables or degrees of freedom to the model increases the complexity of the system very rapidly. A classical challenge for the modelers is to find a trade-off between realism in the modeling framework—for capturing the essence of the question—and simplicity—for being able to solve the question with the available tools and knowledge.

4 Further Reading

More extended background on modeling genetic regulatory networks can be found in [25, 49]. Explanations about different analytical tools such as solving differential equations, nullcline analysis, and linear stability analysis can be found in [5, 8, 13, 26, 50, 51, 52].

Acknowledgements

The code used in this manuscript was loosely based on a code written together with Michael Elowitz, Amit Lakhanpal, and Jordi Garcia-Ojalvo. We would like to thank Iftach Nachman and Oren Shaya for their instructive comments of the manuscript. D. S. would like to thank the support of the Community under a Marie Curie European Reintegration Grant and a grant from the Israel Science Foundation (grant 1021/11).

References

1. Andersson ER, Sandberg R, Lendahl U (2011) Notch signaling: simplicity in design, versatility in function. Development 138: 3593–3612
2. Hori K, Sen A, Artavanis-Tsakonas S (2013) Notch signaling at a glance. J Cell Sci 126: 2135–2140
3. Neves J, Abelló G, Petrovic J et al (2013) Patterning and cell fate in the inner ear: a case for Notch in the chicken embryo. Dev Growth Differ 55:96–112
4. Formosa-Jordan P, Ibañes M, Ares S et al (2013) Lateral inhibition and neurogenesis: novel aspects in motion. Int J Dev Biol 57:341–350
5. Collier JR, Monk NA, Maini P et al (1996) Pattern formation by lateral inhibition with feedback: a mathematical model of delta-notch intercellular signalling. J Theor Biol 183:429–446
6. Owen M, Sherratt J (1998) Mathematical modelling of juxtacrine cell signalling. Math Biosci 153:125–150
7. Meir E, von Dassow G, Munro E et al (2002) Robustness, flexibility, and the role of lateral inhibition in the neurogenic network. Curr Biol 12:778–786
8. Webb SD, Owen MR (2004) Oscillations and patterns in spatially discrete models for developmental intercellular signalling. J Math Biol 48:444–476
9. Webb S, Owen M (2004) Intra-membrane ligand diffusion and cell shape modulate juxtacrine patterning. J Theor Biol 230:99–117
10. Podgorski GJ, Bansal M, Flann NS (2007) Regular mosaic pattern development: a study of the interplay between lateral inhibition, apoptosis and differential adhesion. Theor Biol Med Model 4:43
11. Buceta J, Herranz H, Canela-Xandri O et al (2007) Robustness and stability of the gene regulatory network involved in DV boundary formation in the Drosophila wing. PLoS One 2:e602
12. Morelli LG, Ares S, Herrgen L et al (2009) Delayed coupling theory of vertebrate segmentation. HFSP J 3:55–66
13. Formosa-Jordan P, Ibañes M (2009) Diffusible ligand and lateral inhibition dynamics for pattern formation. J Stat Mech Theor E 2009, P03019
14. Momiji H, Monk N (2009) Oscillatory Notch-pathway activity in a delay model of neuronal differentiation. Phys Rev E 80:21930
15. Barad O, Rosin D, Hornstein E et al (2010) Error minimization in lateral inhibition circuits. Sci Signal 3:ra51
16. Cohen M, Georgiou M, Stevenson NL et al (2010) Dynamic filopodia transmit intermittent Delta-Notch signaling to drive pattern refinement during lateral inhibition. Dev Cell 19:78–89

17. Sprinzak D, Lakhanpal A, Lebon L et al (2010) Cis-interactions between Notch and Delta generate mutually exclusive signalling states. Nature 465:86–90
18. Sprinzak D, Lakhanpal A, Lebon L et al (2011) Mutual inactivation of Notch receptors and ligands facilitates developmental patterning. PLoS Comp Biol 7:e1002069
19. Shaya O, Sprinzak D (2011) From Notch signaling to fine-grained patterning: modeling meets experiments. Curr Opin Genet Dev 21:732–739
20. Stamataki D, Holder M, Hodgetts C et al (2011) Delta1 expression, cell cycle exit, and commitment to a specific secretory fate coincide within a few hours in the mouse intestinal stem cell system. PLoS One 6:e24484
21. Yaron A, Sprinzak D (2012) The cis side of juxtacrine signaling: a new role in the development of the nervous system. Trends Neurosci 35:230–239
22. Matsuda M, Koga M, Nishida E et al (2012) Synthetic signal propagation through direct cell-cell interaction. Sci Signal 5:ra31
23. Formosa-Jordan P, Ibañes M, Ares S et al (2012) Regulation of neuronal differentiation at the neurogenic wavefront. Development 139:2321–2329
24. Okubo Y, Sugawara T, Abe-Koduka N et al (2012) Lfng regulates the synchronized oscillation of the mouse segmentation clock via trans-repression of Notch signalling. Nat Commun 3:1141
25. Alon U (2007) An introduction to systems biology: design principles of biological circuits. CRC Press, Boca Raton, FL
26. Murray JD (2002) Mathematical biology. Springer, New York
27. Cross M, Greenside H (2009) Pattern formation and dynamics in nonequilibrium systems. Cambridge University Press, Cambridge
28. Sakamoto K, Ohara O, Takagi M et al (2002) Intracellular cell-autonomous association of Notch and its ligands: a novel mechanism of Notch signal modification. Dev Biol 241: 313–326
29. Fiuza U, Klein T, Martinez Arias A et al (2010) Mechanisms of ligand-mediated inhibition in Notch signaling activity in Drosophila. Dev Dyn 239:798–805
30. Formosa-Jordan P, Ibañes M (2014) Competition in Notch Signaling with Cis Enriches Cell Fate Decisions. PLoS One 9:e95744
31. D'Souza B, Miyamoto A, Weinmaster G (2008) The many facets of Notch ligands. Oncogene 27:5148–5167
32. Guruharsha K, Kankel M, Artavanis-Tsakonas S (2012) The Notch signalling system: recent insights into the complexity of a conserved pathway. Nat Rev Genet 13:654–666
33. Bollenbach T, Pantazis P, Kicheva A et al (2008) Precision of the Dpp gradient. Development 135:1137–1146
34. O'Dea RD, King JR (2011) Multiscale analysis of pattern formation via intercellular signalling. Math Biosci 231:172–185
35. O'Dea RD, King JR (2012) Continuum limits of pattern formation in hexagonal-cell monolayers. J Math Biol 64:579–610
36. Pennington MW, Lubensky DK (2010) Switch and template pattern formation in a discrete reaction-diffusion system inspired by the Drosophila eye. Eur Phys J E Soft Matter 33:129–148
37. Lubensky DK, Pennington MW, Shraiman BI et al (2011) A dynamical model of ommatidial crystal formation. Proc Natl Acad Sci U S A 108:11145–11150
38. Simakov DS, Pismen LM (2013) Discrete model of periodic pattern formation through a combined autocrine-juxtacrine cell signaling. Phys Biol 10:046001
39. Tanemura M, Honda H, Yoshida A (1991) Distribution of differentiated cells in a cell sheet under the lateral inhibition rule of differentiation. J Theor Biol 153:287–300
40. Eglen SJ, Willshaw DJ (2002) Influence of cell fate mechanisms upon retinal mosaic formation: a modelling study. Development 129: 5399–5408
41. Cohen M, Baum B, Miodownik M (2011) The importance of structured noise in the generation of self-organizing tissue patterns through contact-mediated cell-cell signalling. J R Soc Interface 8:787–798
42. Eldar A, Elowitz MB (2010) Functional roles for noise in genetic circuits. Nature 467: 167–173
43. Gardiner C (2004) Handbook of stochastic methods: for physics, chemistry & the natural sciences, vol 13, Series in synergetics. Springer, New York
44. Rudge T, Burrage K (2008) Effects of intrinsic and extrinsic noise can accelerate juxtacrine pattern formation. Bull Math Biol 70:971–991
45. Lewis J (2003) Autoinhibition with transcriptional delay: a simple mechanism for the zebrafish somitogenesis oscillator. Curr Biol 13:1398–1408
46. Agrawal S, Archer C, Schaffer DV et al (2009) Computational models of the Notch network elucidate mechanisms of context-dependent signaling. PLoS Comput Biol 5:e1000390
47. Petrovic J, Formosa-Jordan P, Luna-Escalante JC et al (2014) Ligand-dependent Notch signaling strength orchestrates lateral induction

and lateral inhibition in the developing inner ear. Development 141:2313–2324

48. Schröter C, Ares S, Morelli LG et al (2012) Topology and dynamics of the zebrafish segmentation clock core circuit. PLoS Biol 10:e1001364
49. Garcia-Ojalvo J (2011) Physical approaches to the dynamics of genetic circuits: a tutorial. Contemp Phys 52:439–464
50. Plahte E (2001) Pattern formation in discrete cell lattices. J Math Biol 43:411–445
51. Strogatz S (2001) Nonlinear dynamics and chaos: with applications to physics, biology, chemistry and engineering. Westview Press, Cambridge
52. Formosa-Jordan P (2013) Pattern formation through lateral inhibition mediated by Notch signaling. University of Barcelona, Barcelona