Comparing Models of Delta-Notch Signalling

Edwin Huras, Riley Wheadon, Ziyi Zhuang

University of British Columbia

April 9, 2025

Introduction

Introduction

The Signalling Pathway

- Ligands bind to a Notch receptor.
- This releases NICD in the cell.
- NICD promotes Notch, inhibits Delta, and promotes Serrate.
- *Cis-inhibition* occurs between molecules from the same cell.

Previous Models

Collier et al. (1996)

- Is a deterministic ODE model.
- Not accurate to the biochemistry.
- Uses computational simulations.
- Looks at multiple 1D/2D domains.

Previous Models

Collier et al. (1996)

- Is a deterministic ODE model.
- Not accurate to the biochemistry.
- Uses computational simulations.
- Looks at multiple 1D/2D domains.

Boareto et al. (2015)

- Is a deterministic ODE model.
- Accurate to the biochemistry.
- Uses bifurcation theory.
- Only analyzes two-cell domains.

Previous Models

Collier et al. (1996)

- Is a deterministic ODE model.
- Not accurate to the biochemistry.
- Uses computational simulations.
- Looks at multiple 1D/2D domains.

Boareto et al. (2015)

- Is a deterministic ODE model.
- Accurate to the biochemistry.
- Uses bifurcation theory.
- Only analyzes two-cell domains.

We investigate how the results of these models change under the **Stochastic Differential Equation** (SDE) and **Agent-Based** formalisms.

Our Assumptions

- All ligands are Delta molecules (so we ignore Serrate).
- Delta molecules only bind to Notch receptors from neighbouring cells.
- Reaction events follow a Poisson point process.
- Notch and Delta production are hill functions of NICD.

Our Chemical Equations

Consider a cell A with neighbour B. The Notch, Delta, and NICD concentrations of cell A are given by N_A, D_A, and I_A respectively (and similarly for cell B).

1. $N_A + D_B \to I_A$: Binding event.
2. $N_A \to \emptyset$: Notch decay.
3. $D_A \to \emptyset$: Delta decay.
4. $I_A \to \emptyset$: NICD decay.
5. $\emptyset \to N_A$: Notch production.
6. $\emptyset \to D_A$: Delta production.

Our Models

In the **Agent-Based** model, we simulate the chemical reactions directly using the *Gillespie Algorithm*. Under our assumptions, this is the most accurate model.

Our Models

In the **Agent-Based** model, we simulate the chemical reactions directly using the *Gillespie Algorithm*. Under our assumptions, this is the most accurate model.

In the **ODE** model, we approximate the reaction kinetics by simulating the agent-based model *in expectation*. To ensure cell differentiation, we apply an *initial perturbation* to all simulations.

Our Models

In the **Agent-Based** model, we simulate the chemical reactions directly using the *Gillespie Algorithm*. Under our assumptions, this is the most accurate model.

In the **ODE** model, we approximate the reaction kinetics by simulating the agent-based model *in expectation*. To ensure cell differentiation, we apply an *initial perturbation* to all simulations.

In the **SDE** model, we reintroduce randomness by adding an independent *Wiener Process* to each equation in the ODE model. This recreates some the stochasticity of the agent-based model.

Derivation

Preliminaries and Assumptions

Reaction	Rate	Description
$N_A + D_B \to I_A$	$k_T N_A D_B$	A Notch receptor from cell A binds to a delta ligand from cell B, triggering the release of NICD in cell A.
$N_A \to \emptyset$	γN_A	A Notch receptor from cell A decays.
$D_A \to \emptyset$	γD_A	A Delta receptor from cell A decays.
$I_A \to \emptyset$	$\gamma_I I_A$	A NICD molecule from cell A decays.
$\emptyset \to N_A$	$H^+(I_A)$	A Notch receptor in cell A is produced
$\emptyset \to D_A$	$H^-(I_A)$	A Delta ligand in cell A is produced

Table 1: A list of possible reactions and their rates in a cell A with neighbour B.

Preliminaries and Assumptions

Since there are 6 possible reactions per cell, a system with k cells will have $6k$ possible reactions to keep track of.

In the following derivations, we focus on tracking a single cell (cell A) for simplicity.

Derivation of Kolmogorov Forward Equation

Let us define

- $P(n, d, i, t)$: the probability of having n Notch receptors, d Delta ligands, and i NICD molecules at time t.
- D_{ext}: the average Delta concentration over all neighbours of cell A.

The KFE describes how $P(n, d, i, t)$ changes over a small time step Δt:

$$
\begin{aligned}
P(n, d, i, t + \Delta t) = P(n, d, i, t) \\
&+ P(n-1, d, i, t) \cdot H^+(i)\Delta t \quad \text{(Notch production)} \\
&+ P(n, d-1, i, t) \cdot H^-(i)\Delta t \quad \text{(Delta production)} \\
&+ P(n+1, d, i, t) \cdot \gamma(n+1)\Delta t \quad \text{(Notch decay)} \\
&+ P(n, d+1, i, t) \cdot \gamma(d+1)\Delta t \quad \text{(Delta decay)} \\
&+ P(n, d, i+1, t) \cdot \gamma_I(i+1)\Delta t \quad \text{(NICD decay)} \\
&+ P(n+1, d, i-1, t) \cdot k_T(n+1)D_{ext}\Delta t \quad \text{(Trans-activation)} \\
&- P(n, d, i, t) \cdot [H^+(i) + H^-(i) + \gamma n + \gamma d + \gamma_I i + k_T n D_{ext}]\Delta t
\end{aligned}
$$

Derivation of Kolmogorov Forward Equation

Rearranging to find the time derivative and taking the limit as $\Delta t \to 0$:

$$\begin{aligned}\frac{\partial P(n,d,i,t)}{\partial t} &= P(n-1,d,i,t)\cdot H^{+}(i) \\ &+ P(n,d-1,i,t)\cdot H^{-}(i) \\ &+ P(n+1,d,i,t)\cdot \gamma(n+1) \\ &+ P(n,d+1,i,t)\cdot \gamma(d+1) \\ &+ P(n,d,i+1,t)\cdot \gamma_I(i+1) \\ &+ P(n+1,d,i-1,t)\cdot k_T(n+1)D_{ext} \\ &- P(n,d,i,t)\cdot [H^{+}(i)+H^{-}(i)+\gamma n+\gamma d+\gamma_I i+k_T n D_{ext}]\end{aligned}$$

Matrix Formulation

The Kolmogorov forward equation can be expressed in matrix form as:

$$\frac{d\vec{P}(t)}{dt} = \mathbf{A}\vec{P}(t)$$

where $\vec{P}(t)$ is a column vector containing the probabilities for all possible states, and $\mathbf{A}$ is the transition rate matrix.
Each entry $A_{i,j}$ represents the transition rate from state i to state j. The diagonal elements $A_{i,i}$ contain the negative sum of all outgoing rates from state i.

Matrix Formulation

Each state in our system is described by (n,d,i), but the matrix form needs a single index.

Matrix Formulation

Each state in our system is described by (n,d,i), but the matrix form needs a single index.

We flatten the 3D state space into a 1D list.

- We encode (n,d,i) into a single index — just like how multidimensional data (e.g., images, simulation grids) are stored in computer memory.

Matrix Formulation

To construct the matrix **A**, we first establish a one-to-one mapping between the three-dimensional state space (n, d, i) and a one-dimensional index k.

We set n_{max}, d_{max}, and i_{max} as large enough bounds for Notch, Delta, and NICD in cell A. These define the size of our 3D "box" of possible states.
Remark: These are not strict limits but chosen large enough to technically capture all relevant dynamics.

Matrix Formulation

For a system with "maximum" values n_{max}, d_{max}, and i_{max}, we define:

$$k(n, d, i) = n \cdot (d_{max} + 1) \cdot (i_{max} + 1) + d \cdot (i_{max} + 1) + i$$

for $0 \leq n \leq n_{max}$, $0 \leq d \leq d_{max}$, $0 \leq i \leq i_{max}$.
The inverse mapping gives:

$$n(k) = \left\lfloor \frac{k}{(d_{max} + 1) \cdot (i_{max} + 1)} \right\rfloor$$
$$d(k) = \left\lfloor \frac{k \bmod ((d_{max} + 1) \cdot (i_{max} + 1))}{i_{max} + 1} \right\rfloor$$
$$i(k) = k \bmod (i_{max} + 1)$$

Matrix Formulation

With this mapping, we can now define the elements of matrix $\mathbf{A}$. Let k and k' be the indices corresponding to states (n, d, i) and (n', d', i') respectively:

$$A_{k,k'} = \begin{cases} H^+(i) & \text{if } (n', d', i') = (n+1, d, i) \quad \text{(Notch production)} \\ H^-(i) & \text{if } (n', d', i') = (n, d+1, i) \quad \text{(Delta production)} \\ \gamma(n) & \text{if } (n', d', i') = (n-1, d, i) \quad \text{(Notch decay)} \\ \gamma(d) & \text{if } (n', d', i') = (n, d-1, i) \quad \text{(Delta decay)} \\ \gamma_I(i) & \text{if } (n', d', i') = (n, d, i-1) \quad \text{(NICD decay)} \\ k_T(n) D_{ext} & \text{if } (n', d', i') = (n-1, d, i+1) \quad \text{(Trans-activation)} \\ -v & \text{if } k' = k \quad \text{(Diagonal terms)} \end{cases}$$

Matrix Formulation

Note that $A_{k,k'}$ represents the transition rate from state k to state k'. The diagonal elements $A_{k,k}$ are the negative sum of all outgoing rates from state k:

$$A_{k,k} = -v_k = -[H^+(i) + H^-(i) + \gamma n + \gamma d + \gamma_I i + k_T n D_{ext} + k_T d N_{ext}]$$

where (n, d, i) is the state corresponding to index k.
Recall: We assume that reaction times follow a Poisson point process.

Matrix Formulation

Note that $A_{k,k'}$ represents the transition rate from state k to state k'. The diagonal elements $A_{k,k}$ are the negative sum of all outgoing rates from state k:

$$A_{k,k} = -v_k = -[H^+(i) + H^-(i) + \gamma n + \gamma d + \gamma_I i + k_T n D_{ext} + k_T d N_{ext}]$$

where (n, d, i) is the state corresponding to index k.
Recall: We assume that reaction times follow a Poisson point process.

The resulting matrix **A** is a sparse matrix with the following properties:

- Dimension: $(n_{max}+1)\cdot(d_{max}+1)\cdot(i_{max}+1) \times (n_{max}+1)\cdot(d_{max}+1)\cdot(i_{max}+1)$
- Number of non-zero elements: $\approx 7 \cdot (n_{max}+1)\cdot(d_{max}+1)\cdot(i_{max}+1)$

Matrix Formulation

The corresponding transition matrix **A** would be:

$$\mathbf{A} = \begin{pmatrix} -v_{000} & \alpha_{000,001} & \alpha_{000,010} & \alpha_{000,011} & \alpha_{000,100} & \alpha_{000,101} & \alpha_{000,110} & \alpha_{000,111} \\ \alpha_{001,000} & -v_{001} & \alpha_{001,010} & \alpha_{001,011} & \alpha_{001,100} & \alpha_{001,101} & \alpha_{001,110} & \alpha_{001,111} \\ \alpha_{010,000} & \alpha_{010,001} & -v_{010} & \alpha_{010,011} & \alpha_{010,100} & \alpha_{010,101} & \alpha_{010,110} & \alpha_{010,111} \\ \alpha_{011,000} & \alpha_{011,001} & \alpha_{011,010} & -v_{011} & \alpha_{011,100} & \alpha_{011,101} & \alpha_{011,110} & \alpha_{011,111} \\ \alpha_{100,000} & \alpha_{100,001} & \alpha_{100,010} & \alpha_{100,011} & -v_{100} & \alpha_{100,101} & \alpha_{100,110} & \alpha_{100,111} \\ \alpha_{101,000} & \alpha_{101,001} & \alpha_{101,010} & \alpha_{101,011} & \alpha_{101,100} & -v_{101} & \alpha_{101,110} & \alpha_{101,111} \\ \alpha_{110,000} & \alpha_{110,001} & \alpha_{110,010} & \alpha_{110,011} & \alpha_{110,100} & \alpha_{110,101} & -v_{110} & \alpha_{110,111} \\ \alpha_{111,000} & \alpha_{111,001} & \alpha_{111,010} & \alpha_{111,011} & \alpha_{111,100} & \alpha_{111,101} & \alpha_{111,110} & -v_{111} \end{pmatrix}$$

where:

$$\alpha_{ndi,n'd'i'} = \begin{cases} H^+(i) + H^-(i) + \gamma n + \gamma d + \gamma_I i + k_T n D_{ext} + k_T d N_{ext} & \text{if } (n,d,i) \to (n',d',i') \\ 0 & \text{otherwise} \end{cases}$$

Remarks

Let us consider

- A one-cell system: $P(n, d, i, t + \Delta t)$
  A k-cell system: $P(n_1, d_1, i_1, n_2, d_2, i_2, ..., n_k, d_k, i_k, t + \Delta t)$
- D_{ext} (the average Delta concentration over all neighbours of cell A) can vary depending on the choice of domain.

Remarks

Let us consider

- A one-cell system: $P(n, d, i, t + \Delta t)$
  A k-cell system: $P(n_1, d_1, i_1, n_2, d_2, i_2, ..., n_k, d_k, i_k, t + \Delta t)$
- D_{ext} (the average Delta concentration over all neighbours of cell A) can vary depending on the choice of domain.

The complexity of a single cell highlights the need for numerical methods to model interactions between cells.

Results

Comparing Two-Cell Models

Noise in the SDE Model

Stability Analysis

We tested model stability by perturbing 5 parameters: N_m, D_m, K_T, γ, and γ_I.

- N_m, D_m are the maximum rates of Notch/Delta production.
- K_T is the binding rate.
- γ, γ_I are the Notch/Delta and NICD decay rates.

Stability Analysis

We tested model stability by perturbing 5 parameters: N_m, D_m, K_T, γ, and γ_I.

- N_m, D_m are the maximum rates of Notch/Delta production.
- K_T is the binding rate.
- γ, γ_I are the Notch/Delta and NICD decay rates.

Due to computational limitations, we investigated the 2-dimensional subspaces of our 5-dimensional parameter space using the following algorithm:

- Run the two-cell model on a 25×25 grid of points spanning 2 OOMs.
- Take the convex hull of the points at which the cells differentiated.

Stability Analysis

Simulations on Linear Domains

Simulations on Linear Domains

Patterns on Linear Domains

Simulations on Hexagonal Domains

Patterns on Hexagonal Domains

Patterns on Hexagonal Domains

Patterns on Hexagonal Domains

Patterns on Hexagonal Domains

Patterns on Hexagonal Domains