

# BE 159: Signal Transduction and Mechanics in Morphogenesis

Justin Bois

Caltech

Winter, 2018

© 2018 Justin Bois.

This work is licensed under a [Creative Commons Attribution License CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/).

## 3 Wnt signaling

In this lecture, we will discuss methods for modeling biochemical networks using mass action kinetics with the example of **Wnt signaling** as our motivation.

### 3.1 Introduction to Wnt signaling

The Wnt (pronounced “wint”) signaling pathway is central in many developmental processes. To see how central it is, you might want to visit [the Wnt homepage](#), run by Roel Nusse’s lab at Stanford, which details the components of the pathway as well as a wealth of links to other information.

The history of the discovery of the Wnt family of proteins highlights its importance in development. In their Nobel Prize-winning work published in 1980, Nüsslein-Volhard and Wieschaus discovered several genes that are central to development in *Drosophila*. One of these was a segment polarity gene Wingless (Wg). The gene was so named because of its phenotype: wingless adult flies, so the gene has downstream effects past regulation of segment polarity. A couple years later, Nusse and Varmus discovered a gene in mice where mutations caused breast cancer, which they named integration 1, or int1. It was later discovered that int1 was highly conserved across species, including *Drosophila*, and that it was part of the same family as Wg. Going forward, this family of genes was referred to as Wnt, a combination of Wg and int.

During development, as we have mentioned in class, neighboring cells need to communicate to each other for differentiation. Beyond that, they need to sense their environment; e.g., they need to make changes to gene expression levels depending on external morphogen concentrations. In order to accomplish this, the “signal” must cross the cell membrane.

The Wnt pathway, shown in Fig. 2 is one major signaling pathway for accomplishing this. The transmembrane proteins Frizzled and LRP (lipoprotein receptor-related protein) are Wnt’s binding partners. When unbound to Wnt, these proteins do not interfere with the destruction cycle of  $\beta$ -catenin, an important transcription factor (more on  $\beta$ -catenin soon). At the center of this destruction cycle is a complex of axin and adenomatous polyposis coli (APC), the latter so named because in humans it was found to be a colorectal tumor suppressor. This complex is commonly referred to as the axis complex. It recruits casein kinase 1 (CK1) and glycogen synthase kinase 3 (GSK-3), which phosphorylate  $\beta$ -catenin. The phosphorylated  $\beta$ -catenin is then targeted by  $\beta$ -TrCP, which promotes polyubiquitination of the phosphorylated  $\beta$ -catenin, which is then degraded by the proteasome.

When Wnt is present outside of the cell membrane, it binds to Fizzled and LRP,

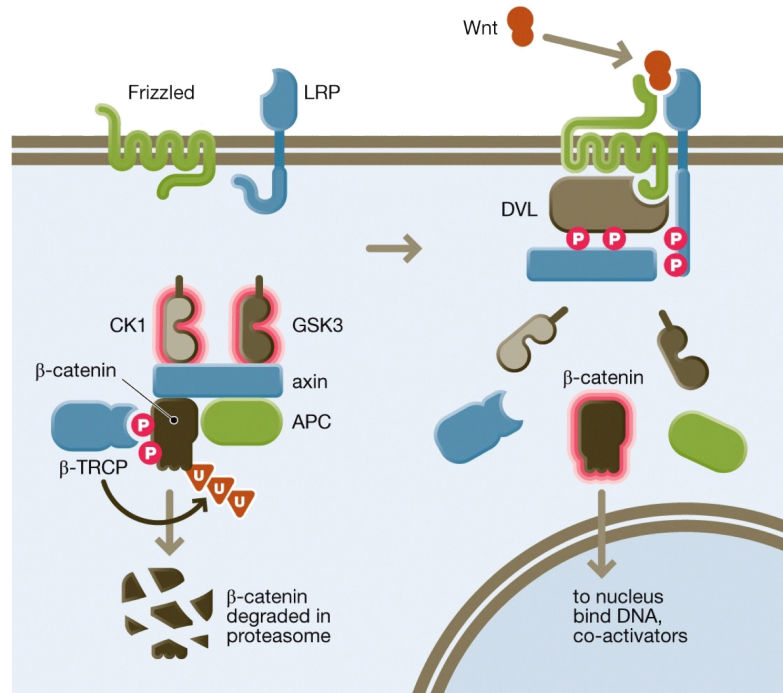


Figure 8.15 Cell Signaling (© Garland Science 2015)

Figure 2: Schematic of the Wnt signaling pathway. Taken from Fig. 8.15 of Lim, Mayer, and Pawson, *Cell Signaling*, Garland Science, 2015.

bringing them together as a heterodimer. In this configuration, Frizzled mediates the phosphorylation and activation of Dishevelled (DVL, a.k.a. Dsh), which then has a strong affinity for axin. Furthermore, the tail of LRP is available for phosphorylation by CK1 and GSK-3. Thus, this activated Frizzled/LRP complex attracts the components of the degradation complex, thereby making them less available for degrading  $\beta$ -catenin. As a result, stable, unphosphorylated  $\beta$ -catenin can enter the nucleus. It then binds its coactivators, e.g., the transcription factor LEF1, and turns on expression of target genes. There are many Wnt-controlled target genes; c-Myc, a multifunctional regulator gene with roles in cellular transformation, is an example.

### 3.2 A more detailed look at Wnt signaling

In the Goentoro, et al. paper we are reading this week, we take a more detailed look of Wnt signaling beyond the cartoon in Fig. 2. The model is based on the work in Lee, et al. from the very first issue of *PLoS Biology*. Their schematic of Wnt signaling is shown in Fig. 3. They have labeled protein-protein interactions with arrows, each one identified with a number, with dashed arrows meaning interactions that are mediated through other proteins. Importantly, they have labeled subprocesses within this spaghetti-looking network to give it clarity. The destruction core cycle

of  $\beta$ -catenin cycles along, provided the equilibrium described by reactions 4 and 5 is unperturbed. The presence of a Wnt molecule affects this equilibrium by activating Dishevelled, which affects the reaction 4/5 equilibrium by breaking down the inactive APC/Axin, GSK-3 complex.

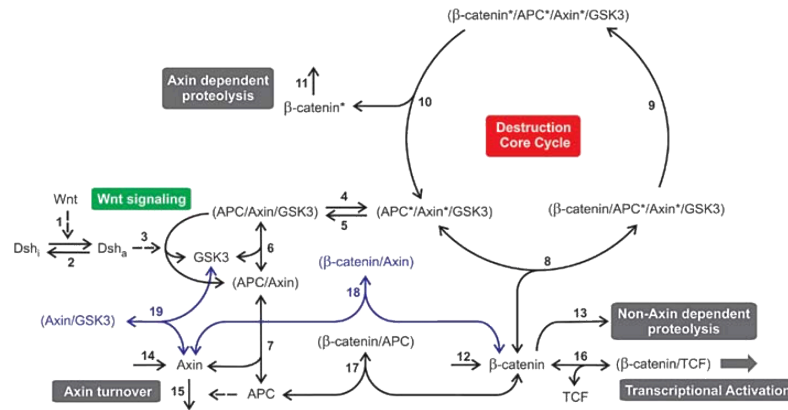


Figure 3: A more detailed model for Wnt signaling. Two headed arrows indicate reversible reactions and one-headed indicate irreversible reactions. Dashed arrows indicate reactions that have other mediators of the reactions. From Lee, et al., *PLoS Biology*, 1, 116–132, 2003.

This is a complicated picture. Our goal is to mathematize this picture using the principle of mass action we talked about in the last lecture, get measured or estimated values for the parameters in the dynamical equations, and compute how changes in Wnt levels affect transcriptional activation.

### 3.3 Mathematizing the cartoon

As is often done in the study of signal transduction networks, **mass action kinetics** are used to model the dynamics. Recalling from last lecture, the rate of a chemical reaction is proportional to the product of the concentrations of the chemical species involved. The constant of proportionality is called the rate constant. Importantly, mass action kinetics do not consider individual reactant molecules, only concentrations of them. Bear in mind also when mass action is valid based on the assumptions we made when deriving it. When the number of reactants are small, or indeed their production is inherently stochastic, as in bursty gene expression, we should instead use stochastic simulation. Because we are not taking into account spatial arrangements of the molecules in our mass action treatment, we are implicitly making a **well-mixed assumption**, meaning that the concentrations are spatially homogeneous, or at least effectively so. Clearly, phosphorylated Dishevelled is not uniformly distributed in space, since it localizes to Frizzled/LRP on the membrane. Nonetheless, we assume that the dynamics of diffusion and spatial organization are fast com-

pared to the chemical kinetics, so we neglect the spatial distribution of molecules. (In future studies, we will not neglect diffusion, to interesting consequences.)

Despite all of these caveats, mass action kinetics seem to be unreasonably effective at describing measured dynamics and making testable predictions. We will therefore employ them in mathematizing the cartoon of the Wnt signaling pathway.

Lee and coworkers write dynamical equations for the entirety of the cartoon, making simplifying assumptions along the way. For demonstration purposes, we will mathematize only the  $\beta$ -catenin destruction core cycle with  $\beta$ -catenin input and phosphorylated  $\beta$ -catenin output. I.e., will will disconnect it from the reversible phosphorylation of APC (reactions 4 and 5 in Fig. 3). Note that reactions 4 and 5 are obviously crucial for getting the full dynamics of Wnt signaling.

In writing the dynamical equations, we do as Lee, et al. and assign numbers for the complexes, since “( $\beta$ -catenin\*/APC\*/Axin\*/GSK3)” is a bit big for a subscript!

number	species
3	APC*/Axin*/GSK-3
8	$\beta$ -catenin/APC*/Axin*/GSK-3
9	$\beta$ -catenin*/APC*/Axin*/GSK-3
10	$\beta$ -catenin*
11	$\beta$ -catenin

Now we can write down the differential equations using mass action.

$$\frac{dc_3}{dt} = -k_8c_3c_{11} + k_{-8}c_8 + k_{10}c_9, \quad (3.1)$$

$$\frac{dc_8}{dt} = k_8c_3c_{11} - k_{-8}c_8 - k_9c_8, \quad (3.2)$$

$$\frac{dc_9}{dt} = k_9c_8 - k_{10}c_9, \quad (3.3)$$

$$\frac{dc_{10}}{dt} = k_{10}c_9 - k_{11}c_{10}, \quad (3.4)$$

$$\frac{dc_{11}}{dt} = k_{12} - k_8c_3c_{11} + k_{-8}c_8. \quad (3.5)$$

We see that

$$\frac{dc_3}{dt} + \frac{dc_8}{dt} + \frac{dc_9}{dt} = 0, \quad (3.6)$$

which implies that the quantity  $c_3 + c_8 + c_9$  is conserved. This makes sense, since this is the total amount of APC/Axin/GSK-3 present. We will call this conserved quantity  $c_A$ .

### 3.3.1 The unique steady state

We can solve for the steady state of this system of ODEs by setting the time derivatives equal to zero and solving. We can subtract equation (3.1) from equation (3.5) and solve to get that  $c_9 = k_{12}/k_{10}$  at steady state. Then, using equations (3.2) and (3.4), we get  $c_8 = k_{12}/k_9$  and  $c_{10} = k_{12}/k_{11}$  at steady state. We then find that at steady state

$$c_3 = c_A - c_8 - c_9 = c_A - \frac{k_{12}}{k_9} - \frac{k_{12}}{k_8}. \quad (3.7)$$

We finally can solve for  $c_{11}$  at steady state to get

$$c_{11} = \frac{k_{12}}{k_8} \left(1 - \frac{k_{\cdot 8}}{k_9}\right) \left(c_A - \frac{k_{12}}{k_9} - \frac{k_{12}}{k_8}\right)^{-1}. \quad (3.8)$$

So, we have found a unique steady state. That the steady state exists and is unique is a useful piece of information in and of itself. We have also found that the steady state values of all species depend on the production rate of  $\beta$ -catenin,  $k_{12}$ .

### 3.3.2 Numerical solution

A system of linear ODEs is easily solved numerically using `scipy.integrate.odeint()`. In solving the ODEs, we take an initial condition of no  $\beta$ -catenin at all in the system, starting only with Axin complex. The total concentration of Axin complex is conserved, with a level of 50 nM, as given in the Lee, et al. paper. We take all other parameters as those reported in the paper as well. The two parameters that are not reported there are  $k_{\cdot 8}$  and  $k_{12}$ . (Actually,  $k_8$  is not reported either, but  $K_{d,8} = k_{\cdot 8}/k_8$  is reported.)

It is easiest to see the effects of varying  $k_{12}$  and  $k_{\cdot 8}$  using interactive plotting. To do this, I used the Python code below, which can be downloaded [here](#). To get the interactive plot, input

```
bokeh serve betacat_destruction_cycle.py
```

at the command line. Then, open a web browser and go to the address

[http://localhost:5006/betacat\\_destruction\\_cycle](http://localhost:5006/betacat_destruction_cycle).

You will need to have a working Python 3 distribution with recent versions of NumPy, SciPy, Pandas, and Bokeh installed.

A sample of the plot is shown in Fig. 4. In moving the sliders in the interactive plot, we see that  $k_{12}$  serves to set the scale of  $\beta$ -cat and  $\beta$ -cat\* concentrations. Varying  $k_8$  sets the total amount of  $\beta$ -catenin. Interestingly, for these parameter values, the concentrations of all Axin-associated complexes is essentially constant. We could make this approximation in the dynamics and get simplified equations for the kinetics.

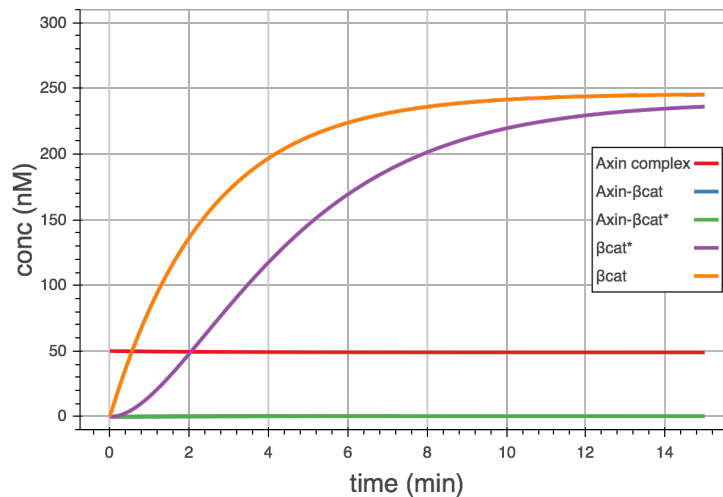


Figure 4: The dynamics of the major species in the  $\beta$ -catenin destruction cycle with all parameters as given in Lee, et al., and  $k_8 = 1 \text{ min}^{-1}$  and  $k_{12} = 100 \text{ (nM-min)}^{-1}$ .

```
1 """
2 To do interactive plotting, run:
3
4     bokeh serve betacat_destruction_cycle.py
5
6 on the command line. Then, point a web browser to
7 http://localhost:5006/betacat_destruction_cycle.
8 """
9
10 import numpy as np
11 import pandas as pd
12 import scipy.integrate
13
14 import bokeh.core.properties
15 import bokeh.io
```

```

16 import bokeh.models.widgets
17 import bokeh.palettes
18 import bokeh.plotting
19
20 def dcdt(c, t, k8, km8, k9, k10, k11, k12):
21     """
22     Time derivative of concentrations.
23     c = (c3, c8, c9, c10, c11)
24     """
25     # Unpack concentrations and parameters
26     c3, c8, c9, c10, c11 = c
27
28     # Build derivatives
29     deriv = np.empty(5)
30     deriv[0] = -k8*c3*c11 + km8*c8 + k10*c9
31     deriv[1] = k8*c3*c11 - (km8 + k9)*c8
32     deriv[2] = k9*c8 - k10*c9
33     deriv[3] = k10*c9 - k11*c10
34     deriv[4] = -k8*c3*c11 + km8*c8 + k12
35
36     return deriv
37
38 # Key for names
39 names = ['Axin complex', 'Axin $\beta$ -cat', 'Axin $\beta$ -cat*', ' $\beta$ cat*', ' $\beta$ cat']
40
41 # Specify colors
42 colors = ['#e41a1c', '#377eb8', '#4daf4a', '#984ea3', '#ff7f00']
43
44 # Define known parameters from Lee, et al, PLoS Biology, 2003
45 c_A = 50          # nM (given by fixed GSK-3 concentration)
46 k9 = 206          # 1/min
47 k10 = 206         # 1/min
48 k11 = 0.417      # 1/min
49 Kd8 = 120         # nM
50
51 # Unknown parameters
52 log10_km8 = 0     # log10(1/min)
53 k12 = 100         # nM/min
54
55 # k8 determined from Kd8 and km8
56 km8 = 10**log10_km8
57 k8 = km8 / Kd8   # 1/nM-min
58
59 # Initial conditions
60 c0 = np.array([c_A, 0, 0, 0, 0])
61
62 # Set up time points and solve
63 t = np.linspace(0, 15, 400)
64 c = scipy.integrate.odeint(dcdt, c0, t, args=(k8, km8, k9, k10, k11,
65     k12))
66 # Store in a DataFrame for convenience in plotting

```



```

67 df = pd.DataFrame(data=c, columns=names)
68 df['t'] = t
69
70 # Data source
71 source = bokeh.models.ColumnDataSource(data=df)
72
73 # Set up the figure
74 p = bokeh.plotting.Figure(plot_width=650, plot_height=450,
75                             x_axis_label='time (min)',
76                             y_axis_label='conc (nM)',
77                             y_range=[-10,310],
78                             border_fill_alpha=0, background_fill_alpha
79                             =0)
80
81 # Add glyphs
82 for i, name in enumerate(names):
83     p.line(x='t', y=name, source=source, line_width=3, color=colors[i
84           ],
85           legend=bokeh.core.properties.value(name))
86
87 # Place legend
88 p.legend.location = 'top_left'
89
90 # Set up widgets
91 k12_val = bokeh.models.Slider(title='k12 [1/nm-min]',
92                                value=100.0,
93                                start=20.0,
94                                end=100.0,
95                                step=1.0)
96
97 log10_km8_val = bokeh.models.Slider(title='log10 km8 [log10(1/min)]',
98                                       value=0.0,
99                                       start=-2.0,
100                                      end=4.0,
101                                      step=0.1)
102
103 # Set up callbacks
104 def update_data(attrname, old, new):
105
106     # Compute k8
107     log10_km8 = log10_km8_val.value
108     km8 = 10**log10_km8
109     k8 = km8 / Kd8
110
111     # Generate the new curve
112     c = scipy.integrate.odeint(dcdt, c0, t,
113                                 args=(k8, km8, k9, k10, k11, k12_val.value))
114     df = pd.DataFrame(data=c, columns=names)
115     df['t'] = t
116
117     # Re-source
118     source.data = dict(df)

```

```
117 # Change values upon activating slider
118 for widget in [k12_val, log10_km8_val]:
119     widget.on_change('value', update_data)
120
121 # Set up layouts and add to document
122 inputs = bokeh.models.layouts.WidgetBox(children=[k12_val,
123     log10_km8_val])
124 bokeh.io.curdoc().add_root(bokeh.models.Row(children=[inputs, p],
125     width=800))
```

betacat\_destruction\_cycle.py