Notch signalling in an array of many cells: a model for the PSM, with noisy regulation of each *her1* and *her7* gene copy.

- **List the molecules involved and give each one an index number**

```
In[2204]:= moltypes = {"g1her1", "g2her1", "g1her7", "g2her7",
               "mher1", "mher7", "pher1", "pher7", "mdelta", "pdelta", "pnicd"};
           (* Prefixes g1 and g2, respectively, denote the maternal and paternal gene copies,
           each of which may be in a blocked or active state
            according to which regulatory proteins are bound to its
            promoter. Prefix m denotes mRNA. Prefix p denotes protein. *)
           nmols = Length[moltypes];
           ig1her1 = 1; (* Give the molecules index numbers according
            to the sequence in which they appear in the  moltypes list. *)
           ig2her1 = 2;
           ig1her7 = 3;
           ig2her7 = 4;
           imher1 = 5;
           imher7 = 6;
           ipher1 = 7;
           ipher7 = 8;
           imdelta = 9;
           ipdelta = 10;
           ipnicd = 11;
```

- **Set the timespan of simulation and the number of elementary time-steps corresponding to one minute**

```
In[2216]:= timestep = 1; (* Do not change this. (The dynamical equations assume timestep=1). To
            adjust the number of minutes corresponding to a computer timestep,
           change the line below, specifying the length of a minute in computer timesteps *)
           minute = 4 * timestep;
           tfinal = Round[1000 minute];
```

- **Define the size of the system (n1 cells x n2 cells).**

```
In[2219]:= n1 = 10;
           n2 = 10;
```

- **Define the geometry of the system and assign an index number to each cell.
Specify the topology by listing the neighbours of each cell. Choose betwen cyclic and non-cyclic boundary conditions.**

```
In[2221]:= latticevector1 = N[{Sqrt[3], 0}];(* for hexagonal lattice *)
           latticevector2 = N[{Sqrt[3] / 2, 3 / 2}]; (* for hexagonal lattice *)

           addresses = Flatten[Table[{ja, jb}, {ja, 0, n1 - 1}, {jb, 0, n2 - 1}], 1];
           (* list of lattice addresses of the cells in the patch *)

           ncells = Length[addresses];
           index[{ja_, jb_}] := jb + 1 + n2 * ja;
           (*  serial number of cell at lattice address {ja,jb} *)

           cyclicBoundaryConditions = True;
           (* Set to False for non-cyclic boundary conditions, such that cells at
            the edges of the n1 x n2 array have no neighbours beyond the edges *)

           If[cyclicBoundaryConditions,
            neighbouraddresses[jcell_] :=
              ({j1, j2} = addresses[[jcell]]; {{Mod[j1 + 1, n1], j2},
                {Mod[j1 - 1, n1], j2}, {j1, Mod[j2 + 1, n2]}, {j1, Mod[j2 - 1, n2]},
                {Mod[j1 - 1, n1], Mod[j2 + 1, n2]}, {Mod[j1 + 1, n1], Mod[j2 - 1, n2]}}),
            neighbouraddresses[jcell_] := ({j1, j2} = addresses⟦jcell⟧;
              Select[{{j1 + 1, j2}, {j1 - 1, j2}, {j1, j2 + 1}, {j1, j2 - 1}, {j1 - 1, j2 + 1},
                 {j1 + 1, j2 - 1}}, 0 ≤ #1⟦1⟧ ≤ n1 - 1 && 0 ≤ #1⟦2⟧ ≤ n2 - 1 &])];

           Table[neighbourindices[jcell] = Flatten[index /@ neighbouraddresses[jcell]],
             {jcell, 1, ncells}];

           xyposition[jcell_] :=
             latticevector1 * addresses[[jcell, 1]] + latticevector2 * addresses[[jcell, 2]] ;
```

- **Specify default values for the lifetimes, delays, critical concentrations, and rate constants .**

Notation:
In the assignments below,
bm... denotes mRNA degradation rate (i.e. inverse of lifetime)
bp... denotes protein degradation rate (i.e. inverse of lifetime)

Values for delays and lifetimes below are loosely based on Lewis (Current Biol., 2003), Giudicelli et al. (PLoS Biol., 2007) and Ozbudak & Lewis (PLoS Genet., 2008).

```
In[2229]:= bmher1 = bmher7 = .23 / minute;
           bpher1 = bpher7 = .23 / minute;
           bmdelta = .23 / minute;
           bpdelta = .23 / minute;
           bpnicd = .23 / minute;
```

We specify the delays as a set of values *delay[[target, agent]]*, meaning that the rate of change of the "target" molecule at time *t* is determined by the value of the "agent" molecule at time *t - delay[[target,agent]]*. In other words, *delay[[target,agent]]* is the delay from making a change in the quantity of *agent* to obtaining a resultant change in the quantity of *target*.
Delays may be different for actions in cis (same cell) and in trans (from neighbouring cells).

Define the tables of cis and trans delays by first setting all to zero, and then specifying values for those that are non-zero.
First index of *cisdelay[[i,j]]* or *transdelay[[i,j]]* specifies target, and second index specifies regulatory molecule.
Since the program represents time as an integer variable, the delays must be specified as integers.

```
In[2234]:= cisdelay = Table[0, {nmols}, {nmols}] minute;

           cisdelay[[imher1, ig1her1]] = cisdelay[[imher1, ig2her1]] =
              cisdelay[[imher1, ig1her7]] = cisdelay[[imher1, ig2her7]] = Round[7 minute];
           cisdelay[[imher7, ig1her1]] = cisdelay[[imher7, ig2her1]] =
              cisdelay[[imher7, ig1her7]] = cisdelay[[imher7, ig2her7]] = Round[7 minute];
           cisdelay[[ipher1, imher1]] = Round[1.1 minute]; (* was 2.8 min in Lewis 2003;
           corrected according to RD Palmiter, Cell 1975,
           data for ovalbumin synthesis in chick *)
           cisdelay[[ipher7, imher7]] = Round[0.7 minute];
           (* was 1.7 min in Lewis 2003; corrected according to RD Palmiter,
           Cell 1975, data for ovalbumin synthesis in chick *)
           cisdelay[[imdelta, ipher1]] = Round[7 minute];
           cisdelay[[imdelta, ipher7]] = Round[7 minute];
           cisdelay[[ipdelta, imdelta]] = Round[20 minute];

           transdelay = Table[0, {nmols}, {nmols}] minute;

           transdelay[[ipnicd, ipdelta]] = Round[2 minute];

           maxdelay = Max[Table[{cisdelay, transdelay}, {jcell, 1, ncells}, {t, 1, tfinal}]];
```

In[2245]:=

```
hillh1 = 2; (* stoichiometry of Her1 binding to DNA *)
hillh7 = 2;(* stoichiometry of Her7 binding to DNA *)
hilln = 1;(* stoichiometry of NICD binding to DNA *)
pcrith1regg = 100; (* critical concentration of pher1 for
 binding to the her1/7 regulatory locus, in molecules per cell *)
pcrith7regg = 100; (* critical concentration of pher7 for binding
 to the her1/7 regulatory locus, in molecules per cell *)
pcritnregg = 50; (* critical concentration of pnicd for binding
 to the her1/7 regulatory locus, in molecules per cell *)
pcrith1regd = pcrith1regg; (* critical concentration of pher1 for
 inhibition of delta gene expression, in molecules per cell *)
pcrith7regd = pcrith7regg; (* critical concentration of pher7 for
 inhibition of delta gene expression, in molecules per cell *)
pcritdregn = 10 000; (* a large value represents the condition that pdelta
 levels are far below the saturating level for Notch activation *)

koffgh = .5 / minute; (* koffgh is the rate constant for
 dissociation of Her protein from the her1/7 promoter/enhancer *)
(* Caution: Note that the stochastic behaviour will be misleadingly
  represented (exaggerated, in fact) if the computer timestep
  (the value of timestep above) is long compared with the equilibration time
  for the association/dissociation reaction between regulatory protein and DNA,
i.e. long compared with 1/koffgh and/or 1/koffgn. For in that case,
the state of the system will be updated at much less frequent intervals than
 required to follow the rapid fluctuations in the state of the gene. *)
koffgn = 1 / minute; (* koffgn is the rate constant for dissociation
 of NICD protein from the her1/7 promoter/enhancer *)
kmher1 = kmher7 = 16.5 / minute;
(* maximal synthesis rate of mher1/7 per gene copy. *)
kpher1 = kpher7 = 9.2 / minute;
(* kpher1/7 is the rate of synthesis of pher1/7 per molecule of mher1/7. *)
(* was 4.5/min in Lewis 2003; corrected according to RD Palmiter,
Cell 1975, data for ovalbumin synthesis in chick *)
kmdelta = 33. / minute; (* maximal synthesis rate of mdelta. *)
kpdelta =  9.2 / minute;
(* kpdelta is the rate of synthesis of pdelta per molecule of mdelta. *)
(* was 4.5/min in Lewis 2003; corrected according to RD Palmiter,
Cell 1975, data for ovalbumin synthesis in chick *)
kn = 0.1 * pcritdregn / minute; (* kn/pcritdregn is the rate of synthesis of NICD
 per molecule of pdelta when pdelta is well below its critical value *)
g1her1Func = 1; (* Set to 1 for a functional gene copy,
0 for a non-functional gene copy (i.e one that generates no transcripts). *)
g2her1Func = 1;(* Set to 1 for a functional gene copy,
0 for a non-functional gene copy (i.e one that generates no transcripts). *)
g1her7Func = 1; (* Set to 1 for a functional gene copy,
0 for a non-functional gene copy (i.e one that generates no transcripts). *)
g2her7Func = 1;(* Set to 1 for a functional gene copy,
0 for a non-functional gene copy (i.e one that generates no transcripts). *)
pher1Func = 1; (* Set to 1 for functional protein,
0 for functionally null protein *)
pher7Func = 1;(* Set to 1 for functional protein, 0 for functionally null protein *)

hillh6 = 2;(* stoichiometry of Hes6 binding to DNA *)
phes6 = 100; (* Concentration of Hes6 protein in molecules per cell,
assumed a constant in this context *)
pcrith6regg = 100;
pcrith6regd = 100;
```

■ Choose between stochastic and deterministic models.

In[2271]:=

```
stochastic = True;
```

■ For the deterministic case: use the expectation value of the state of activity of each *her1* or *her7* gene, so as to compute the smoothed-out behaviour corresponding to very rapid association/dissociation kinetics for the reaction between regulatory proteins and DNA.

We suppose we have a set of protein complexes, Pc0, Pc1, Pc2,... , which compete with one another to bind to the key regulatory site on DNA:

G + Pc0 <-> GPc0
G + Pc1 <-> GPc1,
etc.
Suppose furthermore that the gene is transcriptionally active in the unbound state and in the state with Pc0 bound, but otherwise is inactive. Then it is easy to show that at chemical equilibrium the expectation value of the level of gene activation is simply
Ng (1 + k0 Pc0) / (1+ k0 Pc0 + k1 Pc1 + k2 Pc2 + ...)
where Ng is the number of gene copies and the ki are binding constants.

■ Specify the dynamical rules for regulation of her1 and her7 to be used in the deterministic case

```
In[2272]:= deterministicRules =
   Hold[
    (
     f0[ig1her1, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
      (1 + (cisconcs⟦ipnicd⟧ / pcritnregg)^hilln) / (1 + (cisconcs⟦ipnicd⟧ / pcritnregg)^hilln +
         (cisconcs⟦ipher1⟧ / pcrith1regg)^hillh1 + (cisconcs⟦ipher7⟧ / pcrith7regg)^hillh7 (phes6 / pcrith6regg)^hillh6);

     f0[ig2her1, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
      f0[ig1her1, jcell, t, cisconcs, rcisconcs, rtransconcs];

     f0[ig1her7, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
      f0[ig1her1, jcell, t, cisconcs, rcisconcs, rtransconcs];
     f0[ig2her7, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
      f0[ig2her1, jcell, t, cisconcs, rcisconcs, rtransconcs];
     (* Here we assume that her1 and her7 within a given copy of the her1/7
      complex are coregulated, i.e. controlled by the same regulatory DNA
      and thus always in the same state of inhibition or activation *)

     f0[imher1, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] := (1 - bmher1)
        cisconcs⟦imher1⟧ + kmher1 (rcisconcs[[ig1her1]] + rcisconcs[[ig2her1]]);
     f0[imher7, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] := (1 - bmher7)
        cisconcs⟦imher7⟧ + kmher7 (rcisconcs[[ig1her7]] + rcisconcs[[ig2her7]]);
    )
   ];
```

■ For the stochastic case: Calculate the transition probabilities for switching of each gene copy between active (no inhibitory protein bound) and inactive (inhibitory protein bound) states. The stochastic nature of these transitions is the source of noise in the model.

Consider one gene copy at a time. Suppose this can exist in any one of three mutually exclusive states: with no regulatory protein bound, with NICD protein bound, or with Her protein bound, where the latter state corresponds to repression (no transcription) and the two former states correspond to active transcription. For simplicity, we assume in the first instance that Her1 protein and Her7 protein bind as multimeric complexes to DNA and that these complexes are functionally equivalent, so that the state of the gene depends simply on the sum of their concentrations, which we refer to as the amount of Her complexes in a generic sense. We also allow that NICD may function as a multimeric complex, not necessarily as a monomer.

Let G denote the gene without protein bound, N denote NICD protein complex, H_c denote Her complex, GN the gene with NICD complex bound, and GH_c the gene with Her complex bound. The various association and dissociation reactions are then:
$G + H_c \to GH_c$
$GH \to G + H_c$
$G + N \to GN$
$GN \to G + N.$
Let p_g, p_{gn}, and p_{gh} denote the probability that the gene is in the free, NICD-bound, or Her-bound state, respec-

tively, and let N and H_c denote the concentrations of the NICD and Her protein complexes in the cell. We assume these concentrations are much greater than 1 molecule per cell, so that the association/dissociation reaction with the gene involves no significant change in the concentration of the free protein. Using a prime to denote rate of change with time, we then have the following equations:

$p_{gh}' = p_g\, H_c\, k_{\mathrm{on}gh} - p_{gh}\, k_{\mathrm{off}gh}$ (1)

$p_{gn}' = p_g\, N\, k_{\mathrm{on}gn} - p_{gn}\, k_{\mathrm{off}gn}$ (2)

$p_g = 1 - p_{gh} - p_{gn}$ (3)

The DNA-binding NICD and Her protein complexes are or may be multimeric. We assume that we are in a low-concentration regime where the concentration of n-mers is proportional to the nth power of the concentration of monomers. Thus if NICD functions as an n_N-mer, we assume N, the concentration of the active complex, is proportional to $[\mathrm{NICD}]^{n_N}$. Likewise, we assume that H_c is proportional to $\alpha\, [\mathrm{Her1}]^{n_{\mathrm{h1}}} + \beta\, [\mathrm{Her7}]^{n_{\mathrm{h7}}}$ where α and β are constants. If, for example, Her1 binds as a homotetramer and Her7 as a dimer of heterodimers with Hes6, this latter term becomes $\alpha\, [\mathrm{Her1}]^4 + \beta'\, [\mathrm{Her7}]^2[\mathrm{Hes6}]^2$.

In the steady state,

$p_{gh} = p_g\, H_c\, k_{\mathrm{on}gh} / k_{\mathrm{off}gh}$

$p_{gn} = p_g\, N\, k_{\mathrm{on}gn} / k_{\mathrm{off}gn}$.

If we define $p_{\mathrm{crit}N}$ as the concentration of N at which $p_g = p_{gn}$ at steady state, and $p_{\mathrm{crit}H}$ as the concentration of H_c at which $p_g = p_{gh}$ at steady state, we have

$k_{\mathrm{on}gn} = k_{\mathrm{off}gn} / p_{\mathrm{crit}N}$

and likewise

$k_{\mathrm{on}gh} = k_{\mathrm{off}gh} / p_{\mathrm{crit}H}$.

Substituting from (3) in (1) and (2), we have

$p_{gh}' = ((1-p_{gh}-p_{gn})\, H_c / p_{\mathrm{crit}H} - p_{gh})\, k_{\mathrm{off}gh}$ (1')

$p_{gn}' = ((1-p_{gh}-p_{gn})\, N / p_{\mathrm{crit}N} - p_{gn})\, k_{\mathrm{off}gn}$ (2')

i.e., putting for short

$H_c / p_{\mathrm{crit}hc\mathrm{reg}g} = a$,

$k_{\mathrm{off}gh} = b$,

$N / p_{\mathrm{crit}N} = c$,

$k_{\mathrm{off}gn} = d$,

we have

$p_{gh}' = b * ((1-p_{gh}-p_{gn})\, a - p_{gh})$

$p_{gn}' = d * ((1-p_{gh}-p_{gn})\, c - p_{gn})$.

We solve these equations for each of the three possible initial conditions $\{p_{gn}[0] = p_{gh}[0] = 0,\ p_{gn}[0] = 1\ \&\ p_{gh}[0] = 0,\ p_{gn}[0] = 0\ \&\ p_{gh}[0] = 1\}$ to obtain the probability, during one timestep of duration t, of each possible type of transition from one state of the gene to another.

```
soln0 = First[FullSimplify[DSolve[{pgh'[t] == b * ((1 - pgh[t] - pgn[t]) a - pgh[t]),
     pgn'[t] == d * ((1 - pgh[t] - pgn[t]) c - pgn[t]), pgh[0] == 0, pgn[0] == 0},
    {pgh[t], pgn[t]}, t]]] (* To save unnecessarily repeating this time-
  consuming calculation every time the program is executed,
 I have performed the calculation once and,
 in a subsequent cell in the cell group,
 have defined soln0 to be equal to the resulting value.  The
   initial (visible) cell of the group is now set as non-
  evaluatable. Likewise for soln1 and soln2, below. *)
```

$$\begin{aligned}
&\Bigl\{\mathrm{pgh}[t]\to\Bigl(a\,e^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\Bigl(-\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}-\\
&\quad\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}+\\
&\quad 2\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;e^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}+\\
&\quad (b\,(1+a+2\,c)-(1+c)\,d)\left(-1+e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}\right)\Bigr)\Bigr)\Bigm/\\
&\Bigl(2\,(1+a+c)\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\Bigr),\ \mathrm{pgn}[t]\to\\
&\Bigl(c\,e^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\Bigl(-\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}-\\
&\quad\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}+\\
&\quad 2\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;e^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}+\\
&\quad (-(1+a)\,b+(1+2\,a+c)\,d)\left(-1+e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}\right)\Bigr)\Bigr)\Bigm/\\
&\Bigl(2\,(1+a+c)\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\Bigr)\Bigr\}
\end{aligned}$$

- **Set soln0 to the form computed above**

In[2273]:= **soln0 =**

$$\begin{aligned}
&\Bigl\{\mathrm{pgh}[t]\to\Bigl(a\,e^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\Bigl(-\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}-\\
&\quad\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}+\\
&\quad 2\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;e^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}+\\
&\quad (b\,(1+a+2\,c)-(1+c)\,d)\left(-1+e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}\right)\Bigr)\Bigr)\Bigm/\\
&\Bigl(2\,(1+a+c)\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\Bigr),\ \mathrm{pgn}[t]\to\\
&\Bigl(c\,e^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\Bigl(-\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}-\\
&\quad\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}+\\
&\quad 2\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;e^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}+\\
&\quad (-(1+a)\,b+(1+2\,a+c)\,d)\left(-1+e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}\right)\Bigr)\Bigr)\Bigm/\\
&\Bigl(2\,(1+a+c)\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\Bigr)\Bigr\};
\end{aligned}$$

```
soln1 =
 First[FullSimplify[DSolve[{pgh'[t] == b * ((1 - pgh[t] - pgn[t]) a - pgh[t]), pgn'[t] == d *
       ((1 - pgh[t] - pgn[t]) c - pgn[t]), pgh[0] == 1, pgn[0] == 0}, {pgh[t], pgn[t]}, t]]]
```

$$
\begin{aligned}
&\Big\{\mathrm{pgh}[t] \to \Bigg(\mathrm{e}^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t} \\
&\quad\Bigg(b\,(-1+a\,(-1+c)-c)\left(-1+\mathrm{e}^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\;t}\right)+ \\
&\quad\quad(1+c)^2\,d\left(-1+\mathrm{e}^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\;t}\right)+ \\
&\quad\quad\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2} \\
&\quad\quad\Bigg(2\,a\,\mathrm{e}^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\right)t}+ \\
&\quad\quad\quad(1+c)\left(1+\mathrm{e}^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\;t}\right)\Bigg)\Bigg)\Bigg)\Bigg/ \\
&\quad\left(2\,(1+a+c)\,\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right),\ \mathrm{pgn}[\\
&\quad t] \to \\
&\quad\Bigg(c\,\mathrm{e}^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\Bigg(-\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}- \\
&\quad\quad\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;\mathrm{e}^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}+ \\
&\quad\quad 2\,\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;\mathrm{e}^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}- \\
&\quad\quad(b+a\,b+d+c\,d)\left(-1+\mathrm{e}^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}\right)\Bigg)\Bigg)\Bigg/ \\
&\quad\left(2\,(1+a+c)\,\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)\Big\}
\end{aligned}
$$

■ Set soln1 to the form computed above

In[2274]:=

$$
\begin{aligned}
&\texttt{soln1} = \Bigg\{\texttt{pgh[t]} \to \Bigg(e^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\\
&\quad\Bigg(b\,(-1+a\,(-1+c)-c)\left(-1+e^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\;t}\right)+\\
&\quad\quad(1+c)^2\,d\left(-1+e^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\;t}\right)+\\
&\quad\quad\sqrt{\left((1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2\right)}\\
&\quad\quad\Bigg(2\,a\,e^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\right)t}+\\
&\quad\quad\quad(1+c)\left(1+e^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\;t}\right)\Bigg)\Bigg)\Bigg)\Bigg/\\
&\quad\left(2\,(1+a+c)\,\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right),\ \texttt{pgn[t]} \to\\
&\quad\Bigg(c\,e^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\Bigg(-\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}-\\
&\quad\quad\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}+\\
&\quad\quad 2\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;e^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}-\\
&\quad\quad(b+a\,b+d+c\,d)\left(-1+e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\;t}\right)\Bigg)\Bigg)\Bigg/\\
&\quad\left(2\,(1+a+c)\,\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)\Bigg\};
\end{aligned}
$$

```
soln2 =
  First[FullSimplify[DSolve[{pgh'[t] == b * ((1 - pgh[t] - pgn[t]) a - pgh[t]), pgn'[t] == d *
        ((1 - pgh[t] - pgn[t]) c - pgn[t]), pgh[0] == 0, pgn[0] == 1}, {pgh[t], pgn[t]}, t]]]
```

$$\Big\{\text{pgh}[t]\to\Big(a\,e^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\Big(-\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}-\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\,e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\,t}+2\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\,e^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}-(b+a\,b+d+c\,d)\left(-1+e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\,t}\right)\Big)\Big)\Big/\left(2\,(1+a+c)\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right),\ \text{pgn}[t]\to\Big(e^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\Big((1+a)^2\,b\left(-1+e^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\,t}\right)+(-1-a+(-1+a)\,c)\,d\left(-1+e^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\,t}\right)+\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\Big(2\,c\,e^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\right)t}+(1+a)\left(1+e^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\,t}\right)\Big)\Big)\Big)\Big/\left(2\,(1+a+c)\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)\Big\}$$

- **Set soln2 to the form computed above**

In[2275]:= **soln2 =**

$$\Big\{\text{pgh}[t]\to\Big(a\,e^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\Big(-\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}-\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\,e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\,t}+2\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\,e^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}-(b+a\,b+d+c\,d)\left(-1+e^{\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\,t}\right)\Big)\Big)\Big/\left(2\,(1+a+c)\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right),\ \text{pgn}[t]\to\Big(e^{-\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)t}\Big((1+a)^2\,b\left(-1+e^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\,t}\right)+(-1-a+(-1+a)\,c)\,d\left(-1+e^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\,t}\right)+\sqrt{\left((1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2\right)}\Big(2\,c\,e^{\frac{1}{2}\left(b+a\,b+d+c\,d+\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\right)t}+(1+a)\left(1+e^{\sqrt{(1+a)^2\,b^2+2\,b\,(-1-a+(-1+a)\,c)\,d+(1+c)^2\,d^2}\,t}\right)\Big)\Big)\Big)\Big/\left(2\,(1+a+c)\sqrt{-4\,b\,(1+a+c)\,d+(b+a\,b+d+c\,d)^2}\right)\Big\};$$

Denoting the free state as the 0 state, the H-bound state as 1, and the N-bound state as 2, we have the following transition probabilities at each timestep (of duration *t*), with the notation $p_{01} = p[0 \to 1]$, etc.,

$p_{00} = 1 - p_{gh}[t] - p_{gn}[t]$ evaluated according to soln0 above

$p_{01} = p_{gh}[t]$ evaluated according to soln0 above

$p_{02} = p_{gn}[t]$ evaluated according to soln0 above

$p_{10} = 1 - p_{gh}[t] - p_{gn}[t]$ evaluated according to soln1 above

$p_{11} = p_{gh}[t]$ evaluated according to soln1 above

$p_{12} = p_{gn}[t]$ evaluated according to soln1 above

$p_{20} = 1 - p_{gh}[t] - p_{gn}[t]$ evaluated according to soln2 above

$p_{21} = p_{gh}[t]$ evaluated according to soln2 above

$p_{22} = p_{gn}[t]$ evaluated according to soln2 above

- **Specify the dynamical rules for regulation of *her1* and *her7* to be used in the stochastic case**

In[2276]:=
```
transitionProbs =
  Hold[
   (
    h1 =.;
    h7 =.;
    n =.;
    abcd = {a → (((h1 / pcrith1regg)^hillh1 + (h7 / pcrith7regg)^hillh7 (phes6 / pcrith6regg)^hillh6)),
      b → koffgh, c → (n / pcritnregg)^hilln, d → koffgn};
    soln0abcd = soln0 /. abcd;
    soln1abcd = soln1 /. abcd;
    soln2abcd = soln2 /. abcd;
    p00[h1_, h7_, n_] = 1 - pgh[t] - pgn[t] /. soln0abcd /. t → timestep;
    p01[h1_, h7_, n_] = pgh[t] /. soln0abcd /. t → timestep;
    p02[h1_, h7_, n_] = pgn[t] /. soln0abcd /. t → timestep;
    p10[h1_, h7_, n_] = 1 - pgh[t] - pgn[t] /. soln1abcd /. t → timestep;
    p11[h1_, h7_, n_] = pgh[t] /. soln1abcd /. t → timestep;
    p12[h1_, h7_, n_] = pgn[t] /. soln1abcd /. t → timestep;
    p20[h1_, h7_, n_] = 1 - pgh[t] - pgn[t] /. soln2abcd /. t → timestep;
    p21[h1_, h7_, n_] = pgh[t] /. soln2abcd /. t → timestep;
    p22[h1_, h7_, n_] = pgn[t] /. soln2abcd /. t → timestep;
   )
  ];
stochasticRules =
  Hold[
   (
    f0[ig1her1, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
     Which[
      cisconcs[[ig1her1]] ⩵ 0, RandomChoice[{
         p00[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]],
         p01[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]], p02[
          cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]]} → {0, 1, 2}],
      cisconcs[[ig1her1]] ⩵ 1, RandomChoice[{
         p10[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]],
         p11[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]], p12[
          cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]]} → {0, 1, 2}],
      cisconcs[[ig1her1]] ⩵ 2, RandomChoice[{
         p20[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]],
         p21[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]],
         p22[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]]} → {0, 1, 2}]
     ];

    f0[ig2her1, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
     Which[
      cisconcs[[ig2her1]] == 0, RandomChoice[{
         p00[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]],
         p01[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]], p02[
          cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]]} → {0, 1, 2}],
      cisconcs[[ig2her1]] == 1, RandomChoice[{
         p10[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]],
         p11[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]], p12[
          cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]]} → {0, 1, 2}],
      cisconcs[[ig2her1]] == 2, RandomChoice[{
         p20[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]],
         p21[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]],
         p22[cisconcs[[ipher1]], cisconcs[[ipher7]], cisconcs[[ipnicd]]]} → {0, 1, 2}]
    ];

   f0[ig1her7, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
    f0[ig1her1, jcell, t, cisconcs, rcisconcs, rtransconcs];
   f0[ig2her7, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
    f0[ig2her1, jcell, t, cisconcs, rcisconcs, rtransconcs];
   (* Here we assume that her1 and her7 within a given copy of the her1/7
    complex are coregulated, i.e. controlled by the same regulatory DNA
    and thus always in the same state of inhibition or activation *)

  f0[imher1, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
    (1 - bmher1) cisconcs⟦imher1⟧ + kmher1 (If[rcisconcs[[ig1her1]] == 1, 0, g1her1Func] +
         If[rcisconcs[[ig2her1]] == 1, 0, g2her1Func]);

   f0[imher7, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
    (1 - bmher7) cisconcs⟦imher7⟧ + kmher7 (If[rcisconcs[[ig1her7]] == 1, 0, g1her7Func] +
         If[rcisconcs[[ig2her7]] == 1, 0, g2her7Func]);
   (* rcisconcs[[ig1her1]]==1 is the case where Her protein is bound to the gene,
   repressing it; the other cases,where the gene is free
    (rcisconcs[[ig1her1]]==0) or has NICD bound (rcisconcs[[ig1her1]]==2),
   are assumed to allow active transcription. In each case,
   the retarded value is used, reflecting the delay from
    initiation of transcription to completion of a transcript. *)
  )
 ];
```

Specify the dynamical equations to be actually used

For each kind of molecule, f0 specifies its concentration at the next time point as a function of the currently acting concentrations of the various types of molecules in the same cell (cisconcs) and in the neighbouring cells (transconcs). These “currently acting concentrations” are in general the values that were present at some earlier times, corresponding to delays in the control system, denoted by a prefix r (for retarded). However, sometimes - in particular when a molecule directly regulates its own synthesis, but with a delay - we may need to have f0 depend on both the current value of a concentration (cisconcs) and on its retarded value (rcisconcs). When f0 is called later in the program, it will be with the suitably delayed values of the concentrations as arguments. Note that the program allows for f0 to be different in different cells and at different times.

The program allows for the dynamical rules to be position-dependent (variable from cell to cell) and/or time-dependent.

In[2278]:=

```
If[stochastic == True, ReleaseHold[stochasticRules], ReleaseHold[deterministicRules]];

f0[ipher1, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
  (1 - bpher1) cisconcs[[ipher1]] + kpher1 rcisconcs[[imher1]] pher1Func;
f0[ipher7, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
  (1 - bpher7) cisconcs[[ipher7]] + kpher7 rcisconcs[[imher7]] pher7Func;

f0[imdelta, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
```

$$(1-\mathtt{bmdelta})\ \mathtt{cisconcs}[\![\mathtt{imdelta}]\!] + \mathtt{kmdelta} \Big/ \left(1+\left(\frac{\mathtt{rcisconcs}[\![\mathtt{ipher1}]\!]}{\mathtt{pcrith1regd}}\right)^{\mathtt{hillh1}} + \left(\frac{\mathtt{rcisconcs}[\![\mathtt{ipher7}]\!]}{\mathtt{pcrith7regd}}\right)^{\mathtt{hillh7}} \left(\frac{\mathtt{phes6}}{\mathtt{pcrith6regg}}\right)^{\mathtt{hillh6}}\right);$$

```
f0[ipdelta, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
  (1 - bpdelta) cisconcs[[ipdelta]] + kpdelta * rcisconcs[[imdelta]] ;
f0[ipnicd, jcell_, t_, cisconcs_, rcisconcs_, rtransconcs_] :=
```

$$(1-\mathtt{bpnicd})\ \mathtt{cisconcs}[\![\mathtt{ipnicd}]\!] + \frac{\mathtt{kn}\left(\frac{\mathtt{rtransconcs}[\![\mathtt{ipdelta}]\!]}{\mathtt{pcritdregn}}\right)}{1+\left(\frac{\mathtt{rtransconcs}[\![\mathtt{ipdelta}]\!]}{\mathtt{pcritdregn}}\right)};$$

Non-dimensionalization:
We could choose units for the protein and mRNA concentrations so as to make the critical concentrations equal to 1 (or any other value we please), for each of them, for any chosen one of its actions, leaving the other critical concentrations and the degradation rates b and the transcription initiation rates k and the cis- and trans-delays as the parameters to be explored. However, if we wish to describe events in terms of actual numbers of molecules per cell, this non-dimensionalization is not appropriate.

■ Set the starting conditions and the dimensions of the tables of values that describe the system.

fullhistory is an array of values that describes the history of the system fully, specifying the concentration of each molecule at each time point in each cell. Specifically,
fullhistory [[t, jcell, imol]] is the concentration of molecule *imol* in cell *jcell* at timepoint *t*.
fullhistory [[t]] is a snapshot of the state of the system at timepoint *t*.

recenthistory is just that part of *fullhistory* that we need to know in order to compute the next state of the system.
recenthistory [[1]] is a snapshot of the state of the system at a time preceding the present by an amount *maxdelay*;
recenthistory [[maxdelay+1]] is a snapshot of the present state of the system; that is,
recenthistory [[maxdelay+1, jcell, imol]] is the present concentration of the molecule *imol* in cell *jcell*.

In[2284]:=

```
recenthistory0 = Table[
  If [(jm == ig1her1 || jm == ig2her1 || jm == ig1her7 || jm == ig2her7), 0, 1 * RandomReal[]],
  {jt, 1, 1 + maxdelay}, {jcell, 1, ncells}, {jm, 1, nmols}];
fullhistory0 = Table[If[jt > maxdelay + 1, 0, recenthistory0[[jt, jcell, jm]]],
  {jt, 1, tfinal}, {jcell, 1, ncells}, {jm, 1, nmols}];
```

- **Specify how to apply a full series of updates iteratively to obtain the full spatio-temporal history of the system as it develops subject to the chosen molecular controls, up to time tfinal.**

```
In[2285]:= computebehaviour :=
   (
    fullhistory = fullhistory0;
    recenthistory = recenthistory0;

    timetocompute = Timing[
       Do[
        (
         currentCisMols = Table[
            recenthistory[[1 + maxdelay, jcell, mj]],
            {itargetmol, 1, nmols}, {jcell, 1, ncells}, {mj, 1, nmols}
           ];
         (* currentCisMols[[itargetmol,jcell,mj]]  is
          the current concentration of molecule #mj,  in cell #jcell,
         repeated identically  for all values of #itargetmol  *)
         retardedCisMols = Table[
            recenthistory[[1 + maxdelay - cisdelay[[itargetmol, mj]], jcell, mj]],
            {itargetmol, 1, nmols}, {jcell, 1, ncells}, {mj, 1, nmols}
           ];
         (* retardedCisMols[[itargetmol,jcell,mj]]
          is the concentration of molecule #mj,
         evaluated with the appropriate retardation for its current (timepoint t) cis-
          action on target molecule #itargetmol, in cell #jcell  *)
         retardedTransMols = Table[
            recenthistory[[1 + maxdelay - transdelay[[itargetmol, mj]], jcell, mj]],
            {itargetmol, 1, nmols}, {jcell, 1, ncells}, {mj, 1, nmols}
           ];
         (* retardedTransMols[[itargetmol,jcell,mj]]  is the concentration
          of molecule #mj, evaluated  in cell #jcell with the appropriate
           retardation for its current (timepoint t) trans-action on target
           molecule #itargetmol in the neighbours of cell #jcell . *)

         totNbrsRetardedTransMols  = Table[
            Sum[retardedTransMols[[itargetmol, jnbr, mj]],
             {jnbr, neighbourindices[jcell]}],
            {itargetmol, 1, nmols}, {jcell, 1, ncells}, {mj, 1, nmols}
           ];
         (* totNbrsRetardedTransMols[[itargetmol,jcell,mj]] is the concentration of
          molecule #mj, evaluated with the appropriate retardation for its
           current (timepoint t) trans-action on target molecule #itargetmol,
         summed over all the neighbours of cell #jcell.  *)

         newstate = Table[
            f0[imol, jcell, t, currentCisMols[[imol, jcell]],
             retardedCisMols[[imol, jcell]], totNbrsRetardedTransMols[[imol, jcell]]],
            {jcell, 1, ncells} , {imol, 1, nmols}
           ];
         (*  newstate[[jcell,imol]]  is the concentration to be assigned
          to molecule #imol in cell #jcell at the next timepoint  *)

         recenthistory = Append[Drop[recenthistory, 1], newstate];

         fullhistory[[t + 1]] = newstate;
        ),
        {t, maxdelay + 1, tfinal - 1}
       ];
       allcells = Transpose[fullhistory, {3, 1, 2}];
      ][[1]];
   );
```

■ Specify how to work out oscillation period, damping, amplitude, etc, from computed timecourse (for use in deterministic case only)

```
In[2286]:= printOscillationParams[tseries_, jmol_] :=
             (
              m = tseries[[jmol, All]];
              Do[If[m⟦n + 1⟧ < m⟦n⟧ && m⟦n⟧ > m⟦n - 1⟧, (nmaxpenult = nmaxlast;
                 nmaxlast = n; mmaxpenult = mmaxlast; mmaxlast = m⟦n⟧)], {n, 2, tfinal - 1}];
              Do[If[m⟦n + 1⟧ > m⟦n⟧ && m⟦n⟧ < m⟦n - 1⟧, (nminpenult = nminlast; nminlast = n;
                 mminpenult = mminlast; mminlast = m⟦n⟧)], {n, 2, tfinal - 1}];
              period = (nmaxlast - nmaxpenult);
              ampm = mmaxlast - mminlast;
              ampdecfacm = (mmaxlast - mminlast) / (mmaxpenult - mminpenult);
              Print["For molecule ", moltypes[[jmol]],
               " \n   period (in minutes) = ", N[period / minute], "  last peak = ", mmaxlast,
               "  last trough = ", mminlast, "   peak/trough = ", mmaxlast / mminlast,
               "   damping factor = ", (mmaxlast - mminlast) / (mmaxpenult - mminpenult)];
             );
```

■ Specify how to display the results as graphs of time course for each cell and for the mean over all cells

```
In[2287]:= printVals[listParameterNames_] := Print[Table[listParameterNames[[jlistpn]] <>
                 " = " <> ToString[ToExpression[listParameterNames[[jlistpn]]]] <> "    ",
                {jlistpn, 1, Length[listParameterNames]}] // TableForm];
           printCisDelayTable :=
            (cisDelayTable =
              Table[Flatten[{N[cisdelay[[im]] / minute], "  to control   " <> moltypes[[im]]}],
               {im, 1, nmols}];
             Print["\nDelay (in minutes) for controlling molecule in cis\n",
              Style[TableForm[Insert[cisDelayTable, Append[moltypes, " "], 1],
                TableSpacing → {1, 1}], FontSize → 12, FontFamily → "Arial Narrow"]];)
           printTransDelayTable := (transDelayTable = Table[Flatten[{N[transdelay[[im]] / minute],
                 "  to control   " <> moltypes[[im]]}], {im, 1, nmols}];
             Print["\nDelay (in minutes) for controlling molecule in trans\n",
              Style[TableForm[Insert[transDelayTable, Append[moltypes, " "], 1],
                TableSpacing → {1, 1}], FontSize → 12, FontFamily → "Arial Narrow"]];)
           displaytimecourse :=
             (
              scaling = Table[1, {nmols}];
               (* Default - subsequent lines may modify *)
              scaling⟦imher1⟧ = 1 / 40;
              scaling[[ipher1]] = 1 / 1000;
              scaling⟦ipnicd⟧ = 1 / 2000;
              scaling[[ipdelta]] = 1 / 1000;
              scaling[[ig1her1]] = -1;
              scaling[[ig2her1]] = -1;
              scaledAllCells = Table[scaling[[jmol]] * fullhistory[[t, jcell, jmol]],
                {jcell, 1, ncells}, {jmol, 1, nmols}, {t, 1, tfinal}];
              graph[jcell_] := ListLinePlot[scaledAllCells[[jcell, {ig1her1, imher1, ipher1,
                    ipdelta, ipnicd}, All]], PlotStyle → {{RGBColor[1, 0, 0], Thickness[0.002]},
                   {RGBColor[0, 1, 0], Thickness[0.002]}, {RGBColor[0, 0, 0], Thickness[0.002]},
                   {RGBColor[0, 0, 1], Thickness[0.002]}, {RGBColor[1, 0, 1], Thickness[0.002]}},
                PlotRange → {{0, tfinal}, {-2, 5}}, AspectRatio → 0.6, ImageSize → 400,
                PlotLabel → ("\n g1her1 (red), mher1 (green), pher1 (black), pdelta (blue),
                     pnicd (purple) \ntime in minutes\n  cell # " <>
                   ToString[jcell] <> " at " <> ToString[addresses[[jcell]]]),
                Ticks →  {Table[{100 * nt100, 100 * nt100 / minute}, {nt100, 0, tfinal / 100, 5}],
                  Automatic}];
              gt = Table[graph[njcell], {njcell, 1, ncells}];
              Print["To see time course for each cell individually,
                 \nclick on graph window and scroll sideways.
                 \nConcentrations are scaled for convenient display.\nFor g1her1,
                 value 0 means the gene has no regulatory protein bound,\n
                 -1 means it has Her protein bound, -2 means it has NICD bound.
                 \nStates 0 and -2 are transcriptionally active, state -1 is repressed."];
              Print[GraphicsRow[gt]];
              scaledMeanOverCells =
               Table[scaling[[jmol]] * (1 / ncells) Sum[fullhistory[[t, jcell, jmol]],
                  {jcell, 1, ncells}], {jmol, 1, nmols}, {t, 1, tfinal}];
```

```
   Print
    [graphMeanOverCells =
     ListLinePlot[
      scaledMeanOverCells[[{ig1her1, imher1, ipher1, ipdelta, ipnicd}, All]],
      PlotStyle → {{RGBColor[1, 0, 0], Thickness[0.002]}, {RGBColor[0, 1, 0],
         Thickness[0.002]}, {RGBColor[0, 0, 0], Thickness[0.002]},
        {RGBColor[0, 0, 1], Thickness[0.002]}, {RGBColor[1, 0, 1], Thickness[0.002]}},
      PlotRange → {{0, tfinal}, {-2, 5}}, AspectRatio → 0.6,
      ImageSize → 400, PlotLabel →
       ("Mean over all cells\n g1her1 (red), mher1 (green), pher1 (black),
          pdelta (blue), pnicd (purple) \ntime in minutes "),
      Ticks → {Table[{100 * nt100, 100 * nt100 / minute}, {nt100, 0, tfinal / 100, 5}],
        Automatic}]];
   graphFourierMeanOverCells[jmol_] :=
    (
     m = scaledMeanOverCells[[jmol, All]];
     aft = Abs[Fourier[m]];
     maxfreq = .1 / minute;
     ListLinePlot[Take[aft, Round[maxfreq * Length[aft]]],
      PlotRange -> All, DataRange → {0, minute * maxfreq}, AspectRatio → 0.6,
      ImageSize → 400, PlotLabel → "Mean over all cells\n" <> moltypes[[jmol]] <>
        " Fourier transform; amplitude vs frequency in cycles per minute"]
    )
  );
```

■ Specify how to display the honeycomb pattern of cells and its coloring

```
In[2291]:= nucleardiam = .4;
membranethickness = 0.02;
intercellspace = 0.01;
redCytoplasm = imher1;
greenCytoplasm = imher1;
blueCytoplasm = ipdelta;
redNucleus = imher1;
greenNucleus = imher1;
blueNucleus = ipdelta;
colorscaling = Table[1, {nmols}];
(* Default scaling for colour display. Actual desired scaling set in next lines*);
colorscaling[[imher1]] = 20;
colorscaling[[imdelta]] = 10;
colorscaling[[ig1her1]] = .1;
colorscaling[[ipdelta]] = pcritdregn / 10;
colorscaling[[ipnicd]] = pcritnregg;
bkgrndcolor = {1, 1, 1} * 1;
cellColoring[t_] := (celljts = Table[allcells[[jcell, imol, t]] /
       (allcells[[jcell, imol, t]] + colorscaling[[imol]]), {imol, 1, nmols}];
   u = addresses[[jcell, 1]];
   v = addresses[[jcell, 2]];
   membranecolor = {1, 1, 1};
   cytoplasmcolor =
    {0, celljts[[greenCytoplasm]], celljts[[blueCytoplasm]]};
   nucleuscolor =
    {0, celljts[[greenNucleus]], celljts[[blueNucleus]]}
  );

centre[ni_, nj_] := ni * latticevector1 + nj * latticevector2;
hexverts = N[{{-Sqrt[3] / 2, 1 / 2}, {0, 1},
    {Sqrt[3] / 2, 1 / 2}, {Sqrt[3] / 2, -1 / 2}, {0, -1}, {-Sqrt[3] / 2, -1 / 2}}];
translate[vertexlist_, vector_] := Map[Plus[#, vector] &, vertexlist];
membrane[ni_, nj_] :=
  Polygon[translate[(1 - intercellspace) * hexverts, centre[ni, nj]]];
cytoplasm[ni_, nj_] := Polygon[
   translate[(1 - membranethickness - intercellspace) * hexverts, centre[ni, nj]]];
nucleus[ni_, nj_] := Disk[centre[ni, nj], {nucleardiam, nucleardiam}];
cell[ni_, nj_, membranecolor_, cytoplasmcolor_, nucleuscolor_] := Graphics[{
    RGBColor[membranecolor], membrane[ni, nj],
    RGBColor[cytoplasmcolor], cytoplasm[ni, nj],
    RGBColor[nucleuscolor], nucleus[ni, nj]
   }];
```

```
displaySimple[t_] :=
  Show[
   Table[
    (cellColoring[t];
     cell[u, v, membranecolor, cytoplasmcolor, nucleuscolor]
    ),
    {jcell, 1, ncells}
   ],
   Background → Apply[RGBColor, bkgrndcolor],
   (*PlotRange→{{leftmargin,rightmargin},{bottommargin,topmargin}},*)
   AspectRatio → Automatic, PlotLabel → timelabel,
   ImageSize → 50 * {n1, n2}
  ];

displayCyclic[t_] :=
 (horizrepetition = 1;
  vertrepetition = 1;
  jhoriz = Ceiling[horizrepetition + n2 / 2];
  jvert = vertrepetition;
  leftmargin = Norm[latticevector1] * (1 + n1 * n2 / 2);
  rightmargin = Norm[latticevector1] * ((1 + jhoriz) * n1 - 1);
  bottommargin = Norm[latticevector1] * N[Sqrt[3] / 2];
  topmargin = Norm[latticevector1] * N[Sqrt[3] / 2] * ((1 + jvert) * n2 - 1);
  Show[
   Table[
    (cellColoring[t];
     Table[
      cell[u + n1 * jn1, v + n2 * jn2, membranecolor, cytoplasmcolor, nucleuscolor],
      {jn1, 0, jhoriz}, {jn2, 0, jvert}
     ]
    ),
    {jcell, 1, ncells}
   ],
   Background → Apply[RGBColor, bkgrndcolor], PlotRange →
    {{leftmargin, rightmargin}, {bottommargin, topmargin}}, AspectRatio → Automatic,
   PlotRangeClipping → True,
   PlotLabel → timelabel,
   ImageSize → 50 * {n1 * horizrepetition, n2 * vertrepetition}
  ]
 )
```

- **Specify how to generate frames of a movie of the multicellular array**

```
In[2317]:= makemovie[tstartshow_, tinterval_, tendshow_] :=
             Do[
              (
               timelabel = Style["t = " <> ToString[Round[(tf - 1) / minute]] <> " minutes",
                  "Section", FontSize → 14]; (* Here timelabel is defined as time elapsed since
                 first time point in whole history, which therefore has timelabel 0 *)
               Print[displaySimple[tf]];
               Print[displayCyclic[tf]];
              ), {tf, tstartshow, tendshow, tinterval}
             ];

           makemovieRectangle[tstartshow_, tinterval_, tendshow_] :=
             Do[
              (
               timelabel = Style["t = " <> ToString[Round[(tf - 1) / minute]] <> " minutes",
                  "Section", FontSize → 14];
               Print[displayCyclic[tf]];
              ), {tf, tstartshow, tendshow, tinterval}
             ];

           makemovieSimple[tstartshow_, tinterval_, tendshow_] :=
             Do[
              (
               timelabel = Style["t = " <> ToString[Round[(tf - 1) / minute]] <> " minutes",
                  "Section", FontSize → 14];
               Print[displaySimple[tf]];
              ), {tf, tstartshow, tendshow, tinterval}
             ];
```

■ Specify any sets of variant parameters to be explored

Lists of variant values specified here for rate constants, critical concentrations and other parameters appearing in the dynamical equations are to be used in the computation, overriding default values specified earlier.

```
In[2320]:= variedParams =
             {"cisdelay[[ipdelta,imdelta]]/minute", "cisdelay[[imher1,ig1her1]]/minute",
              "cisdelay[[imher7,ig1her7]]/minute", "pcrith1regg", "pcrith7regg",
              "pher1Func", "pher7Func", "seedRandom", "koffgh*minute"};
           variants["cisdelay[[ipdelta,imdelta]]"] = Round[{18} minute];
           variants["cisdelay[[imher1,igher1]]"] = Round[{8} minute];
           variants["cisdelay[[imher7,igher7]]"] = Round[{7} minute];
           variants["pcrith1regg"] = {400};
           variants["pcrith7regg"] = {400};
           variants["pher7Func"] = {1, 0};
           variants["pher1Func"] = {1};
           variants["seedRandom"] = {4};
           variants["koffgh"] = {0.5} / minute;
```

■ Do the computation and display time-course graphs and movie

Values specified here for rate constants, critical concentrations and other parameters appearing in the dynamical equations override default values specified earlier. The Do loop runs over the chosen set of different parameter choices.

```
In[2330]:= Do[
             (
              cisdelay[[ipdelta, imdelta]] = variants["cisdelay[[ipdelta,imdelta]]"][[v1]];
              cisdelay[[imher1, ig1her1]] =
               cisdelay[[imher1, ig2her1]] = cisdelay[[imher1, ig1her7]] =
                 cisdelay[[imher1, ig2her7]] = variants["cisdelay[[imher1,igher1]]"][[v2]];
              cisdelay[[imher7, ig1her1]] = cisdelay[[imher7, ig2her1]] =
                cisdelay[[imher7, ig1her7]] =
                 cisdelay[[imher7, ig2her7]] = variants["cisdelay[[imher7,igher7]]"][[v3]];
              pcrith1regg = pcrith1regd = variants["pcrith1regg"][[v4]];
              pcrith7regg = pcrith7regd = variants["pcrith7regg"][[v9]];
              pher7Func = variants["pher7Func"][[v5]];
              pher1Func = variants["pher1Func"][[v8]];
              seedRandom = variants["seedRandom"][[v6]];
              koffgh = variants["koffgh"][[v7]];

              SeedRandom[seedRandom]; (*Setting SeedRandom[n], where n is any integer,
   SeedRandom[seedRandom]; (*Setting SeedRandom[n], where n is any integer,
   means that the set of random numbers used in the computation subsequent
    to the SeedRandom statement is reproducible,i.e the same each time the
    program is run. To run the program with a different set of random numbers,
   change the value of n. To make every run use a different set of random numbers,
   leave the argument of SeedRandom blank,i.e write simply SeedRandom[]*)
   If[stochastic == True, ReleaseHold[transitionProbs], Null];
   (* This statement has to be here,
   and not in the earlier specification of dynamical rules, because otherwise
    transitionProbs would be computed using default and not variant parameters *)

   Print["\n\n--------------------------------------\nFOR GRAPHS THAT FOLLOW:"];
   printVals[{"moltypes", "minute/timestep", "tfinal/minute", "ncells", "hillh1",
     "hillh7", "hilln", "pcrith1regg", "pcrith7regg", "pcritnregg", "pcrith1regd",
     "pcrith7regd", "pcritdregn", "bmher1*minute", "bpher1*minute", "bmher7*minute",
     "bpher7*minute", "bmdelta*minute", "bpdelta*minute", "bpnicd*minute",
     "kmher1*minute", "kpher1*minute", "kmdelta*minute", "kpdelta*minute", "kn*minute",
     "koffgh*minute", "koffgn*minute", "g1her1Func", "g2her1Func", "pher1Func",
     "pher7Func", "cyclicBoundaryConditions", "seedRandom", "stochastic"}];
   printCisDelayTable;
   printTransDelayTable;
   Print["\n\n"];
   Print[ProgressIndicator[Dynamic[t], {0, tfinal}]];
   computebehaviour;
   printVals[{"timetocompute"}];
   displaytimecourse;
   printVals[variedParams];
   Print["For molecule ",
    moltypes[[imher7]], " (mean over all cells) \nMean over time = ",
    Mean[scaledMeanOverCells[[imher7]]], "   StandardDeviation = ",
    StandardDeviation[scaledMeanOverCells[[imher7]]]];
   If[stochastic == True, Null, printOscillationParams[scaledMeanOverCells, imher7]];
   Print[graphFourierMeanOverCells[imher7]];




   tstartshow = 1; (* for the movie *)
   tendshow = tfinal; (* for the movie *)
   tinterval = 5 minute;
   Print["Colour components of cytoplasm are: \nRed = 0 x " <>
     moltypes[[redCytoplasm]] <> "; Green = " <> moltypes[[greenCytoplasm]] <>
     "; Blue = " <> moltypes[[blueCytoplasm]]];
   Print["Colour components of nucleus are: \nRed = 0 x "  <>
     moltypes[[redNucleus]] <> "; Green = " <> moltypes[[greenNucleus]] <>
     "; Blue = " <> moltypes[[blueNucleus]]];
   makemovieRectangle[tstartshow, tinterval, tendshow]


  ),
  {v1, 1, Length[variants["cisdelay[[ipdelta,imdelta]]"]]},
  {v2, 1, Length[variants["cisdelay[[imher,igher]]"]]},
  {v3, 1, Length[variants["cisdelay[[imher,igher]]"]]},
  {v7, 1, Length[variants["koffgh"]]},
  {v4, 1, Length[variants["pcrith1regg"]]},
  {v9, 1, Length[variants["pcrith7regg"]]},
  {v8, 1, Length[variants["pher1Func"]]},
  {v5, 1, Length[variants["pher7Func"]]},
  {v6, 1, Length[variants["seedRandom"]]}

 ];
```

```
--------------------------------------
FOR GRAPHS THAT FOLLOW:
```

```
moltypes = {g1her1, g2her1, g1her7, g2her7, mher1, mher7, pher1, pher7, mdelta, pdelta,
minute/timestep = 4
tfinal/minute = 1000
ncells = 100
hillh1 = 2
hillh7 = 2
hilln = 1
pcrith1regg = 400
pcrith7regg = 400
pcritnregg = 50
pcrith1regd = 400
pcrith7regd = 400
pcritdregn = 10000
bmher1*minute = 0.23
bpher1*minute = 0.23
bmher7*minute = 0.23
bpher7*minute = 0.23
bmdelta*minute = 0.23
bpdelta*minute = 0.23
bpnicd*minute = 0.23
kmher1*minute = 16.5
kpher1*minute = 9.2
kmdelta*minute = 33.
kpdelta*minute = 9.2
kn*minute = 1000.
koffgh*minute = 0.5
koffgn*minute = 1
g1her1Func = 1
g2her1Func = 1
pher1Func = 1
pher7Func = 1
cyclicBoundaryConditions = True
seedRandom = 4
stochastic = True
```

Delay (in minutes) for controlling molecule in cis

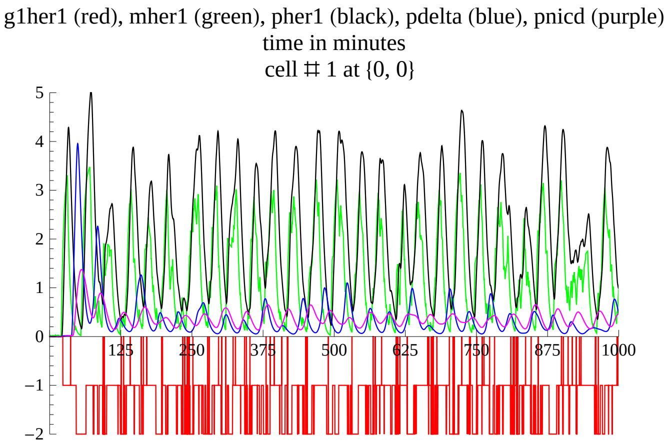
g1her1	g2her1	g1her7	g2her7	mher1	mher7	pher1	pher7	mdelta	pdelta	pnicd		
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g1her1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g2her1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g1her7
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g2her7
8.	8.	8.	8.	0.	0.	0.	0.	0.	0.	0.	to control	mher1
7.	7.	7.	7.	0.	0.	0.	0.	0.	0.	0.	to control	mher7
0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	to control	pher1
0.	0.	0.	0.	0.	0.75	0.	0.	0.	0.	0.	to control	pher7
0.	0.	0.	0.	0.	0.	7.	7.	0.	0.	0.	to control	mdelta
0.	0.	0.	0.	0.	0.	0.	0.	18.	0.	0.	to control	pdelta
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	pnicd

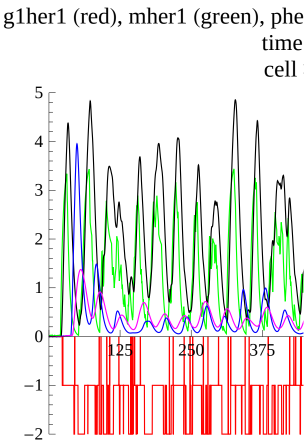
```
Delay (in minutes) for controlling molecule in trans
```

g1her1	g2her1	g1her7	g2her7	mher1	mher7	pher1	pher7	mdelta	pdelta	pnicd		
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g1her1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g2her1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g1her7
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g2her7
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	mher1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	mher7
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	pher1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	pher7
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	mdelta
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	pdelta
0.	0.	0.	0.	0.	0.	0.	0.	0.	2.	0.	to control	pnicd

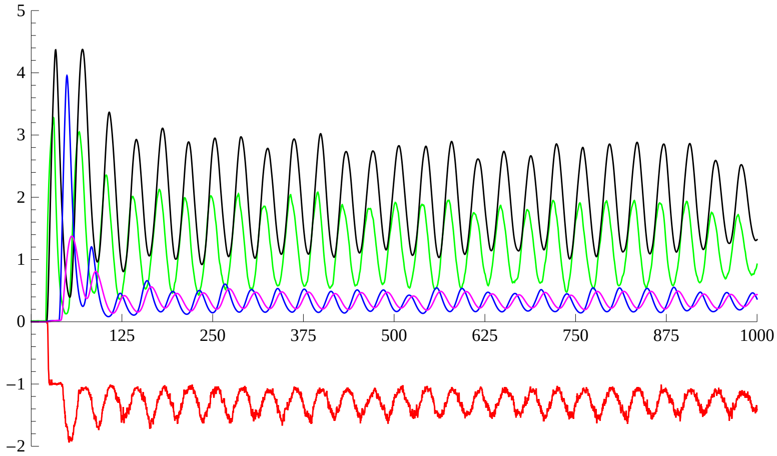
```
timetocompute = 1050.31

To see time course for each cell individually,
click on graph window and scroll sideways.
Concentrations are scaled for convenient display.
For g1her1, value 0 means the gene has no regulatory protein bound,
 -1 means it has Her protein bound, -2 means it has NICD bound.
States 0 and -2 are transcriptionally active, state -1 is repressed.
```





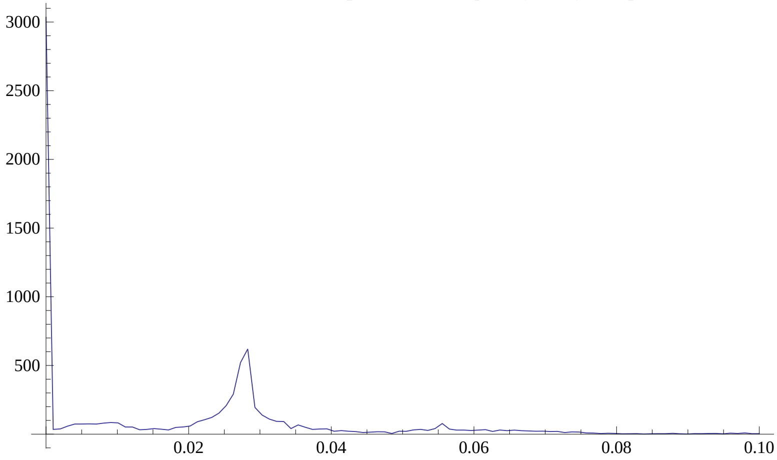
Mean over all cells
g1her1 (red), mher1 (green), pher1 (black), pdelta (blue), pnicd (purple)
time in minutes



```
cisdelay[[ipdelta,imdelta]]/minute = 18
cisdelay[[imher1,ig1her1]]/minute = 8
cisdelay[[imher7,ig1her7]]/minute = 7
pcrith1regg = 400
pcrith7regg = 400
pher1Func = 1
pher7Func = 1
seedRandom = 4
koffgh*minute = 0.5
```

```
For molecule mher7 (mean over all cells)
Mean over time = 47.9828   StandardDeviation = 22.7804
```

Mean over all cells
mher7 Fourier transform; amplitude vs frequency in cycles per minute



Frames of movie are deleted here to keep file size manageable, but are saved as QuickTime movie provided separately.

```
----------------------------------------
FOR GRAPHS THAT FOLLOW:
```

```
moltypes = {g1her1, g2her1, g1her7, g2her7, mher1, mher7, pher1, pher7, mdelta, pdelta,
minute/timestep = 4
tfinal/minute = 1000
ncells = 100
hillh1 = 2
hillh7 = 2
hilln = 1
pcrith1regg = 400
pcrith7regg = 400
pcritnregg = 50
pcrith1regd = 400
pcrith7regd = 400
pcritdregn = 10000
bmher1*minute = 0.23
bpher1*minute = 0.23
bmher7*minute = 0.23
bpher7*minute = 0.23
bmdelta*minute = 0.23
bpdelta*minute = 0.23
bpnicd*minute = 0.23
kmher1*minute = 16.5
kpher1*minute = 9.2
kmdelta*minute = 33.
kpdelta*minute = 9.2
kn*minute = 1000.
koffgh*minute = 0.5
koffgn*minute = 1
g1her1Func = 1
g2her1Func = 1
pher1Func = 1
pher7Func = 0
cyclicBoundaryConditions = True
seedRandom = 4
stochastic = True
```

Delay (in minutes) for controlling molecule in cis

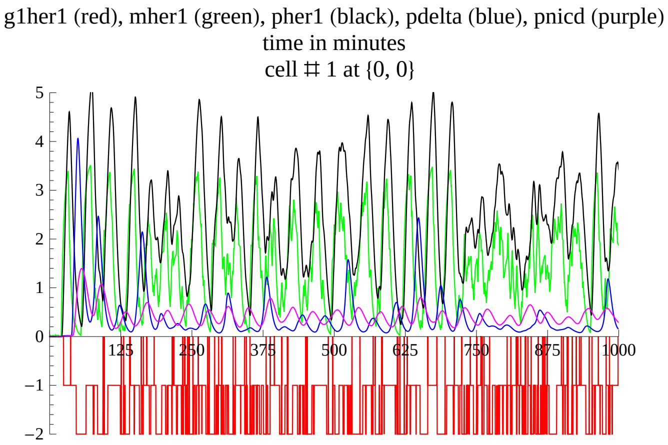
g1her1	g2her1	g1her7	g2her7	mher1	mher7	pher1	pher7	mdelta	pdelta	pnicd		
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g1her1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g2her1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g1her7
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g2her7
8.	8.	8.	8.	0.	0.	0.	0.	0.	0.	0.	to control	mher1
7.	7.	7.	7.	0.	0.	0.	0.	0.	0.	0.	to control	mher7
0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	to control	pher1
0.	0.	0.	0.	0.	0.75	0.	0.	0.	0.	0.	to control	pher7
0.	0.	0.	0.	0.	0.	7.	7.	0.	0.	0.	to control	mdelta
0.	0.	0.	0.	0.	0.	0.	0.	18.	0.	0.	to control	pdelta
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	pnicd

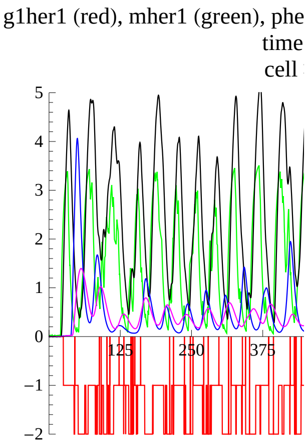
```
Delay (in minutes) for controlling molecule in trans
```

g1her1	g2her1	g1her7	g2her7	mher1	mher7	pher1	pher7	mdelta	pdelta	pnicd		
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g1her1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g2her1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g1her7
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	g2her7
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	mher1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	mher7
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	pher1
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	pher7
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	mdelta
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	to control	pdelta
0.	0.	0.	0.	0.	0.	0.	0.	0.	2.	0.	to control	pnicd

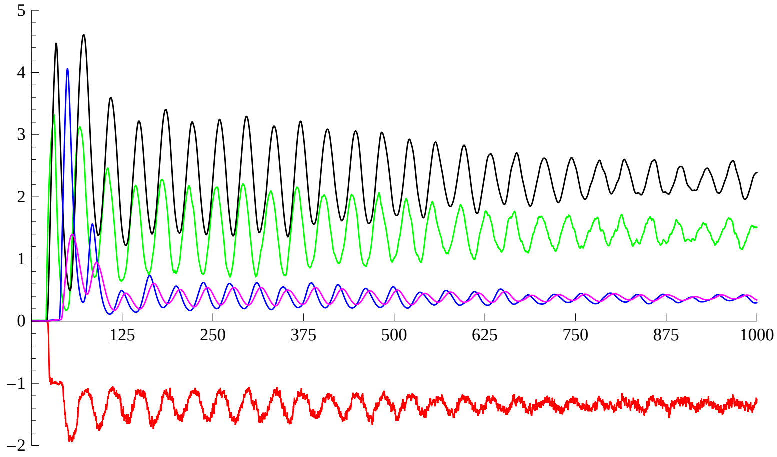
```
timetocompute = 1059.34

To see time course for each cell individually,
click on graph window and scroll sideways.
Concentrations are scaled for convenient display.
For g1her1, value 0 means the gene has no regulatory protein bound,
 -1 means it has Her protein bound, -2 means it has NICD bound.
States 0 and -2 are transcriptionally active, state -1 is repressed.
```





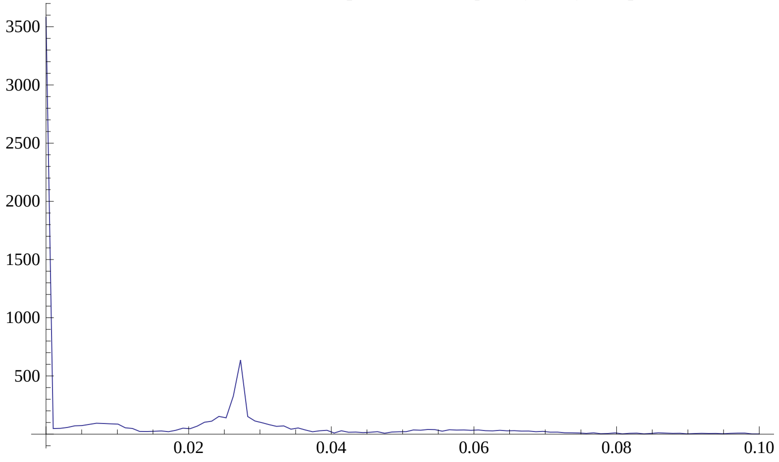
Mean over all cells
g1her1 (red), mher1 (green), pher1 (black), pdelta (blue), pnicd (purple)
time in minutes



```
cisdelay[[ipdelta,imdelta]]/minute = 18
cisdelay[[imher1,ig1her1]]/minute = 8
cisdelay[[imher7,ig1her7]]/minute = 7
pcrith1regg = 400
pcrith7regg = 400
pher1Func = 1
pher7Func = 0
seedRandom = 4
koffgh*minute = 0.5
```

```
For molecule mher7 (mean over all cells)
Mean over time = 56.6304   StandardDeviation = 19.429
```

Mean over all cells
mher7 Fourier transform; amplitude vs frequency in cycles per minute



In[2331]:=

Frames of movie are deleted here to keep file size manageable, but are saved as QuickTime movie provided separately.