Progress in computer algebra based biological modeling languages

Eric Mjolsness

University of California, Irvine

http://emj.ics.uci.edu

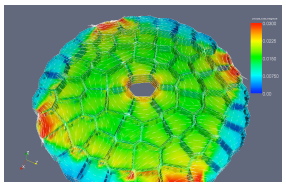
Joint work with Bruce Shapiro, Guy Yosiphon, Henrik Jonsson, Todd Johnson, Elliot Meyerowitz, Victoria Mironova, Vitali Likhoshvai, John Reinitz, David Sharp, and others.

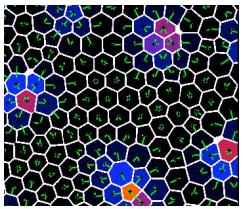
Wolfram Technology Conference

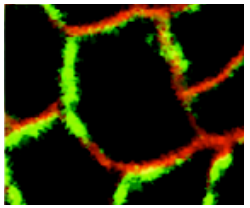
21 October 2013

Cycle of Modeling Scales

with a developmental focus







Development

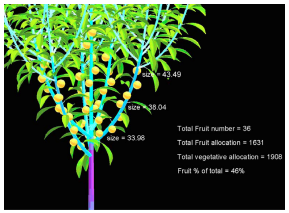
Morphology

Cellular biology

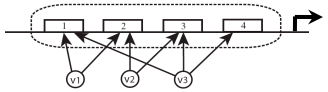
Phenotype

PIN1[i,j]

auxin[i] ⟹ auxin[j]

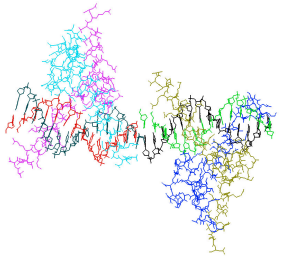


Physiology



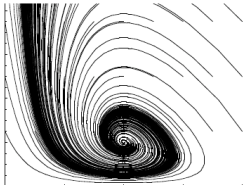
Molecular biology

Genotype

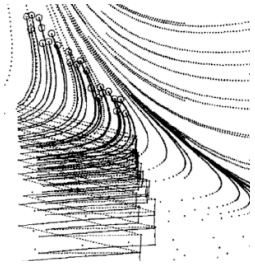


Ecology

Sequence dynamics



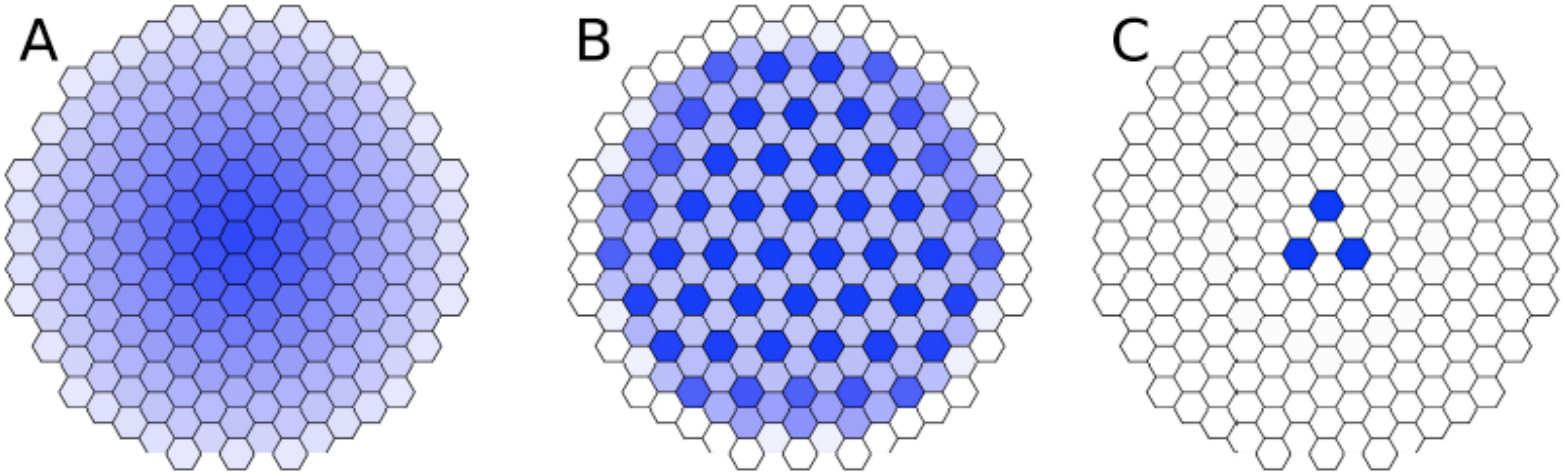
Evolution



Eg: Brusselator in Cellzilla

A diffusible Brusselator is easily implemented in Cellzilla with

$$\{\{\emptyset \rightleftarrows \mathtt{A}, \mathtt{a}, \beta\}, \{2\mathtt{A} + \mathtt{B} \to 3\mathtt{A}, \mathtt{c}\}, \{\mathtt{A} \to \mathtt{B}, \mathtt{b}\},$$
$$\{\mathtt{A} \longrightarrow \mathtt{A}, \mathtt{Diffusion}[D_A, D_A]\}, \{\mathtt{B} \longrightarrow \mathtt{B}, \mathtt{Diffusion}[D_B, D_B]\}\}$$

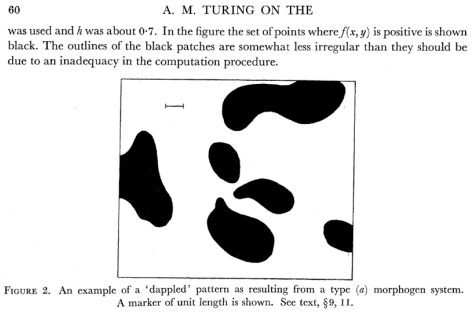


[Shapiro et al., Frontiers in Plant Biophysics and Modeling 4:00408, 2013]

Modeling Morphogenesis

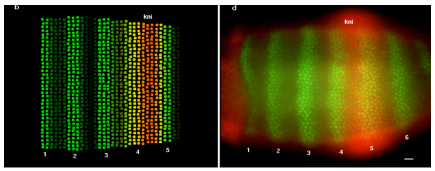
- Reaction-diffusion models
  - A. Turing 1952
    - stripes & spots; floral phyllotactic pattern
    - H. Meinhardt
  - Eg. ANN collective model of GRNs; HCA [BGRS 2006]

$$\frac{dv_a(x)}{dt} = R_a g(h_a + \sum_b T_{ab} v_b(x)) + \Delta_a^{\text{ext}}(x) - \lambda_a v_a(x)$$

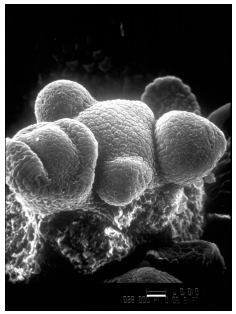


was used and h was about 0·7. In the figure the set of points where $f(x, y)$ is positive is shown black. The outlines of the black patches are somewhat less irregular than they should be due to an inadequacy in the computation procedure.

FIGURE 2. An example of a 'dappled' pattern as resulting from a type (a) morphogen system. A marker of unit length is shown. See text, §9, 11.



- Non-Turing morphogenesis
  - Receptor-ligand spatial interactions
  - Cell polarity vector, tensor dynamics
    - required for floral phyllotactic pattern. Also ...
  - Tissue biomechanics: deformation, growth, topological dynamics



UCI *Mo*

Cellzilla WUS model

/home/mathman/Desktop/Activator.nb

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

Wushel Activator Model

```
xy = Import["geometry.txt","Table"];
sc = surfaceCells /. boundedCellVoronoi[xy, surfaceCells → True];
rates = {ky → 0.2, dy → 0.1, Dy → 0.1, kD → 3.0, tauw → 10, dw → 0.1, hw → 0,
  Twy → -20, Twa → 0.5, a → 0.1, b → 0.2, beta → 0.1, c → 0.1, d → 0.01, Da → 0.1, Db → 1.5};
Clear[L1];
internal[i_] := {
   {Y[i] ↦ W[i], GRN[1/tauw, Twy, 1, hw, sigma]},
   {A[i] ↦ W[i], GRN[1/tauw, Twa, 1, hw, sigma]},
   {W[i] → ∅, dw},
   {∅ ⇄ Y[i], ky * L1[i], dy},
   {∅ ⇄ A[i], a, beta + d * Y[i][t]},
   {2A[i] + B[i] → 3A[i], c},
   {A[i] → B[i], b}
  };
external[i_, j_] := {{Y[i] → Y[j], Dy}, {A[i] → A[j], Da}, {B[i] → B[j], Db}};
SIN =
  createNetwork[xymeristem, internalNetwork → internal, interactionNetwork → external,
   connectionList → "PrunedDelaunay"
  ];
myic[j_] := Join[
   Table[W[i][0] == 0, {i, 1, j}],
   Table[Y[i][0] == 0, {i, 1, j}],
   Table[A[i][0] == 0, {i, 1, j}],
   Table[B[i][0] == 0, {i, 1, j}]
  ];
L1[i_] := If[MemberQ[sc, i], 1, 0];
sim = Timing[run[istn /. rates, {0, 500}, initialConditions → myic[n]]]
```

```
Using Pruned Delaunay Triangulation for Near Neighbors Connections.
2 dimensional data
253 cells
1771 internal reactions
4230 intercellular reactions
0 global reactions
6001 total reactions
```

xCellera

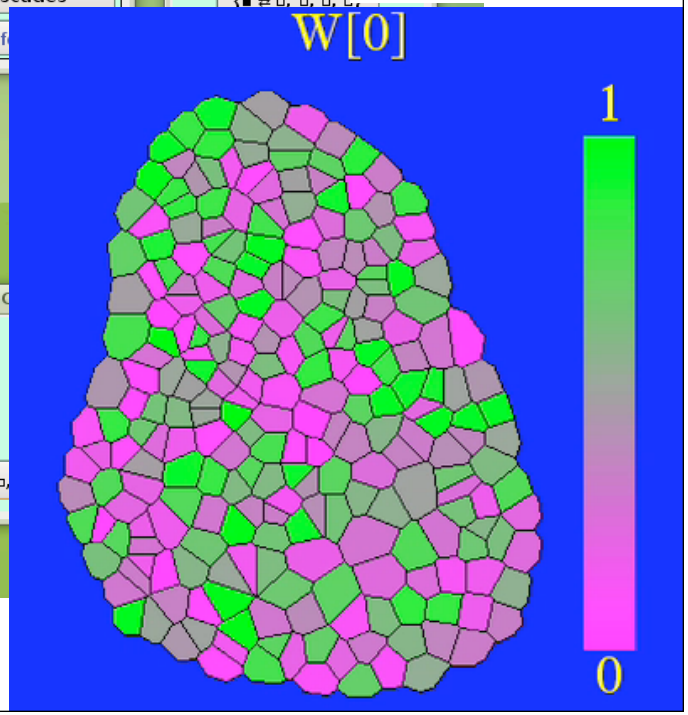
xcellerator

Mass Action
Michaelis – Menten
Hill Functions
GRN Functions
GMWC Reactions
S – Systems
NHCA
User
Arrowzilla
Cascades
More Inf

Massaction.r

xCellerator

Mass Action Rea

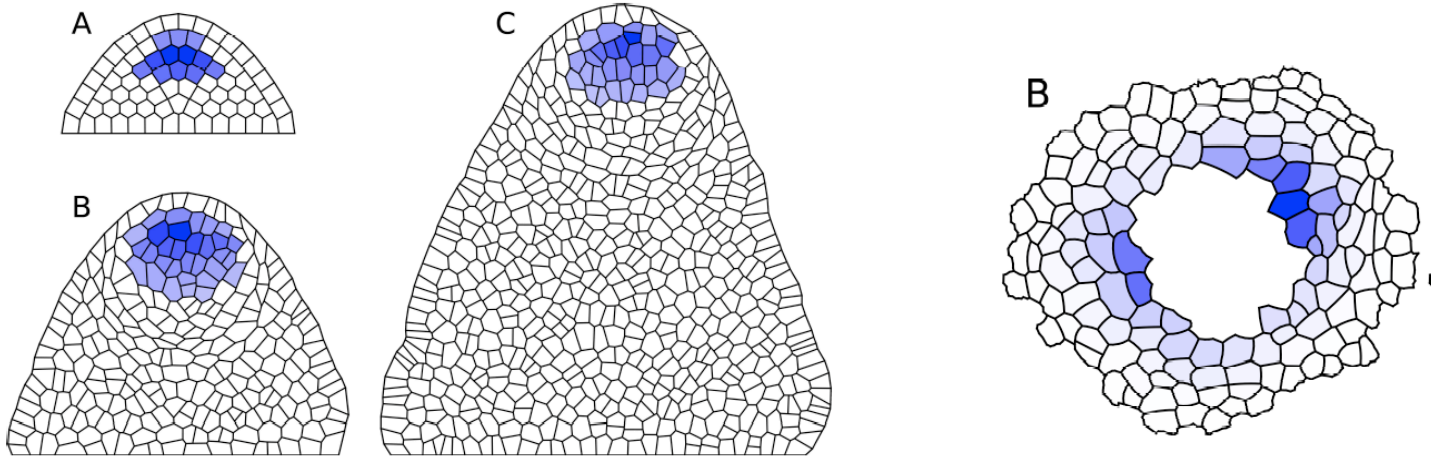


(Bruce Shapiro, Caltech BNMC)

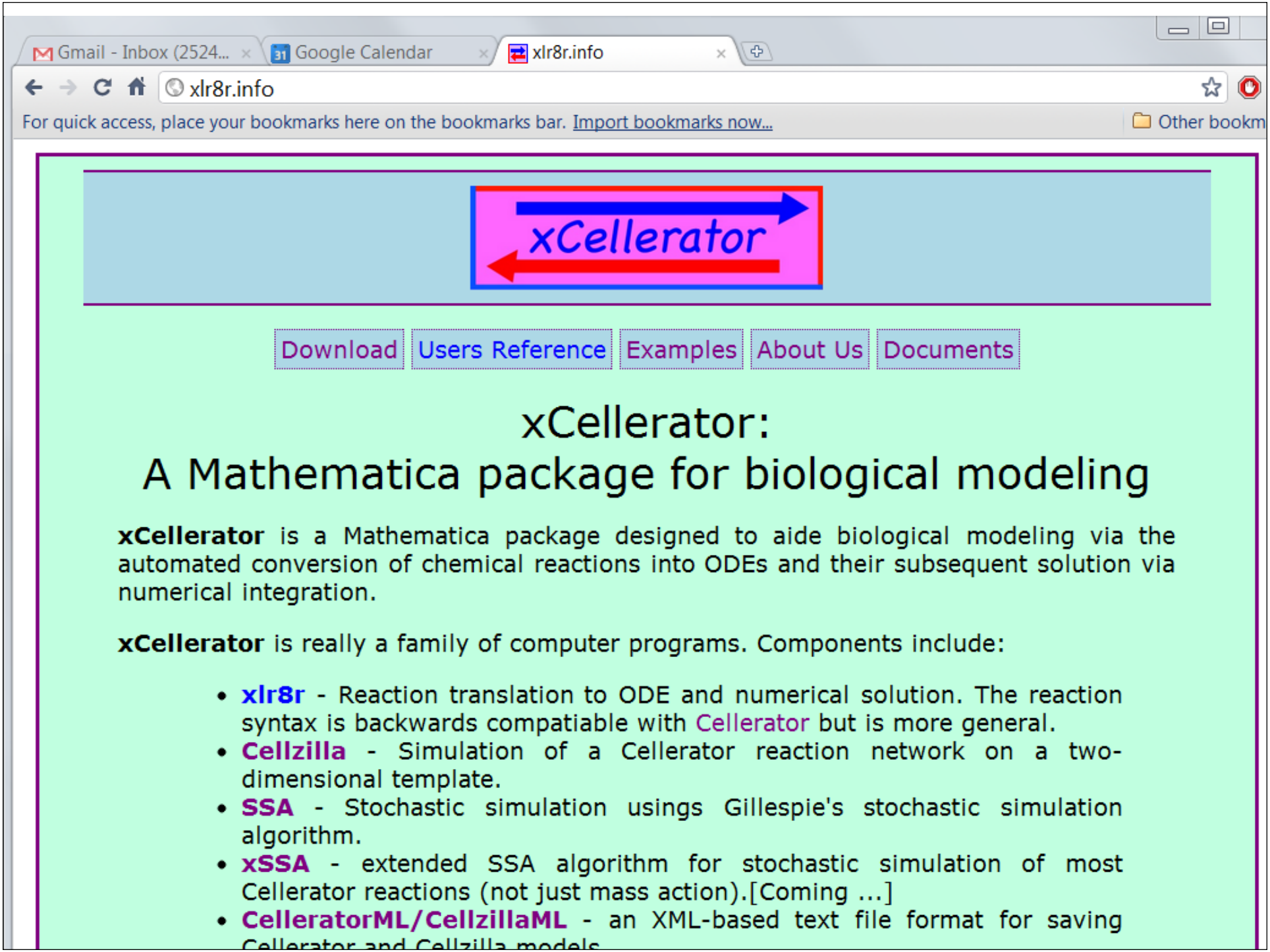
Wuschel model 2013

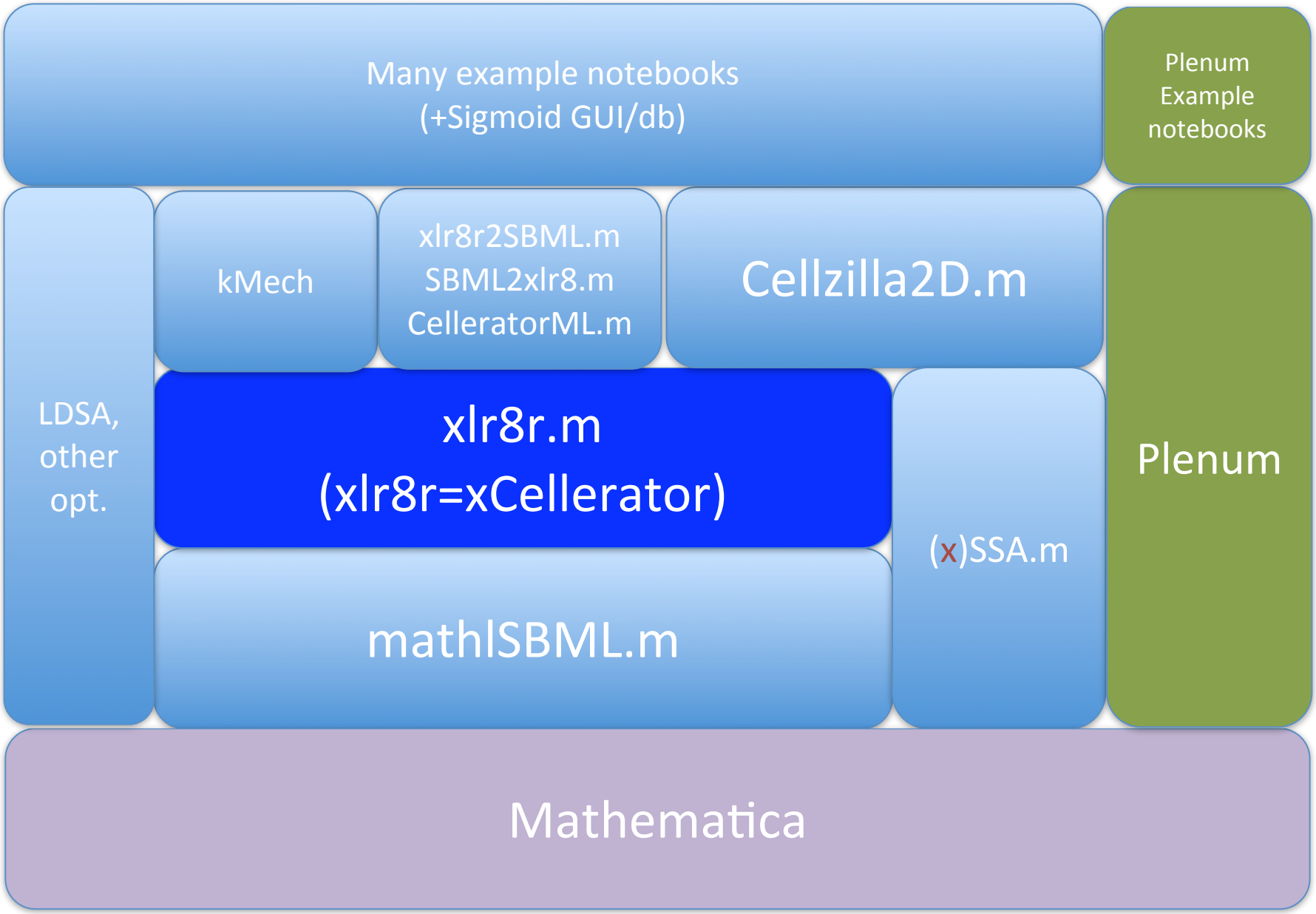
Declarative, with cell growth & division

$$\{\{\emptyset \to \mathtt{U}, \mathtt{k_1TIP[t]}\}, \{\mathtt{U} \to \emptyset, \mathtt{k_2}\}, \{\mathtt{U} \longrightarrow \mathtt{U}, \mathtt{Diffusion[D_U]}\},$$
$$\{\emptyset \to \mathtt{V}, \mathtt{k_3L1[t]}\}, \{\mathtt{V} \to \emptyset, \mathtt{k_4}\}, \{\mathtt{V} \longrightarrow \mathtt{V}, \mathtt{Diffusion[D_V]}\},$$
$$\{\emptyset \rightleftarrows \mathtt{Z}, \mathtt{k_7}, \mathtt{k_8U[t]}\}, \{\mathtt{X} \mapsto \mathtt{V}, \mathtt{GRN[v_V, T_{WV}, 1, h_V]}\},$$
$$\{\{\mathtt{U}, \mathtt{V}, \mathtt{W}\} \mapsto \mathtt{W}, \mathtt{GRN[v_W, \{T_{UW}, T_{VW}, T_{WW}\}, 1, h_W]}\}, \{\mathtt{W} \to \emptyset, \mathtt{k_6Z[t]} + \mathtt{k_9L2[t]}\}$$
$$\{\mathtt{W} \mapsto \mathtt{X}, \mathtt{GRN[v_X, T_{WX}, 1, h_X]}\}, \{\mathtt{X} \to \emptyset, \mathtt{k_5}\}, \{\mathtt{X} \longrightarrow \mathtt{X}, \mathtt{Diffusion[D_X]}\},$$
$$\{\mathtt{cell} \longrightarrow \mathtt{cell}, \mathtt{Grow[GrowthRate}[\mu, \mathtt{f}_\mu], \mathtt{Pressure[P, f_P]}, \mathtt{Spring[k, f_k]}\},$$
$$\{\mathtt{cell} \longrightarrow \mathtt{cell} + \mathtt{cell}, \mathtt{Errera[cell}, \mu, \sigma\}\}$$



[Shapiro et al., Frontiers in Plant Biophysics and Modeling 4:00408, 2013]


Gmail - Inbox (2524...
Google Calendar
xlr8r.info
xlr8r.info
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Other bookm
xCellerator
Download
Users Reference
Examples
About Us
Documents
xCellerator:
A Mathematica package for biological modeling
xCellerator is a Mathematica package designed to aide biological modeling via the automated conversion of chemical reactions into ODEs and their subsequent solution via numerical integration.
xCellerator is really a family of computer programs. Components include:
xlr8r - Reaction translation to ODE and numerical solution. The reaction syntax is backwards compatiable with Cellerator but is more general.
Cellzilla - Simulation of a Cellerator reaction network on a two-dimensional template.
SSA - Stochastic simulation usings Gillespie's stochastic simulation algorithm.
xSSA - extended SSA algorithm for stochastic simulation of most Cellerator reactions (not just mass action).[Coming ...]
CelleratorML/CellzillaML - an XML-based text file format for saving



Feynman diagrams

for stochastic reactions and/or rules

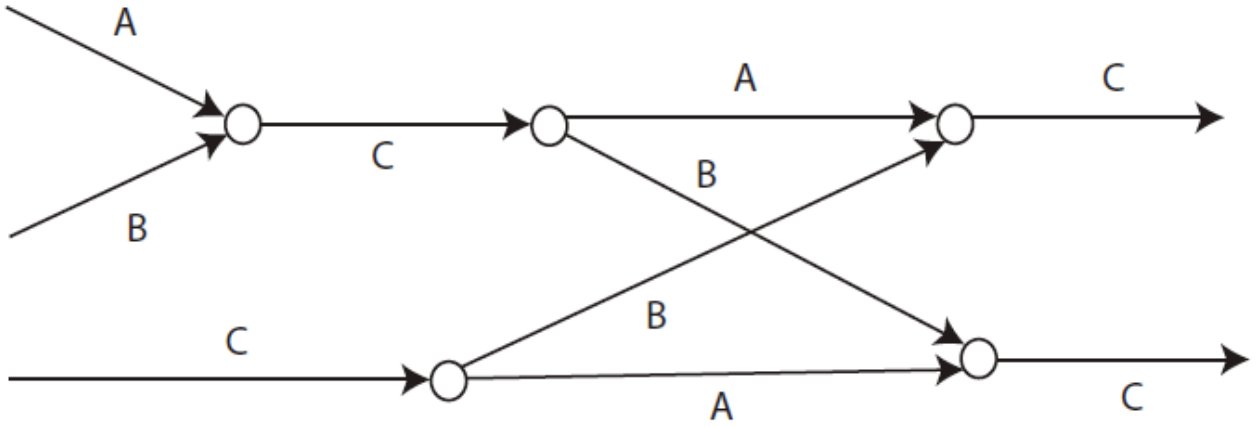


Figure 1. A time history of the reaction $A + B \rightleftharpoons C$. Time flows left to right. Open circles represent reaction events, with the probability factor $\times W_1$. In between, reaction events are unimolecular particle propagators $\exp((t_k - t_{k-1})W_0)$, labeled by arrows and particle names (repeated for clarity). This is a non-spatial version of the Lee model in quantum field theory (cf, for example, [6]).

Phys. Biol. **10** (2013) 035009

Resulting hybrid ODE/stochastic simulation algorithm

factor $\rho^{(r)}(x_{\text{in}},\ x_{\text{out}}) = k^{(r)}(x_{\text{in}})\, p^{(r)}(x_{\text{out}} \mid x_{\text{in}})$;

repeatedly {

initialize SSA propensities as $k^{(r)}(x_{\text{in}})$;

initialize $k^{(\text{total})} = \sum_r k^{(r)}(x_{\text{in}})$;

draw *effective* waiting time $(\Delta T)_{\max}$ from $\exp(-(\Delta T)_{\max})$

solve ODE system, including an extra ODE updating ΔT:

$$\frac{d\Delta T}{dt} = k^{(\text{total})}(t)$$

until $\Delta T = (\Delta T)_{\max}$

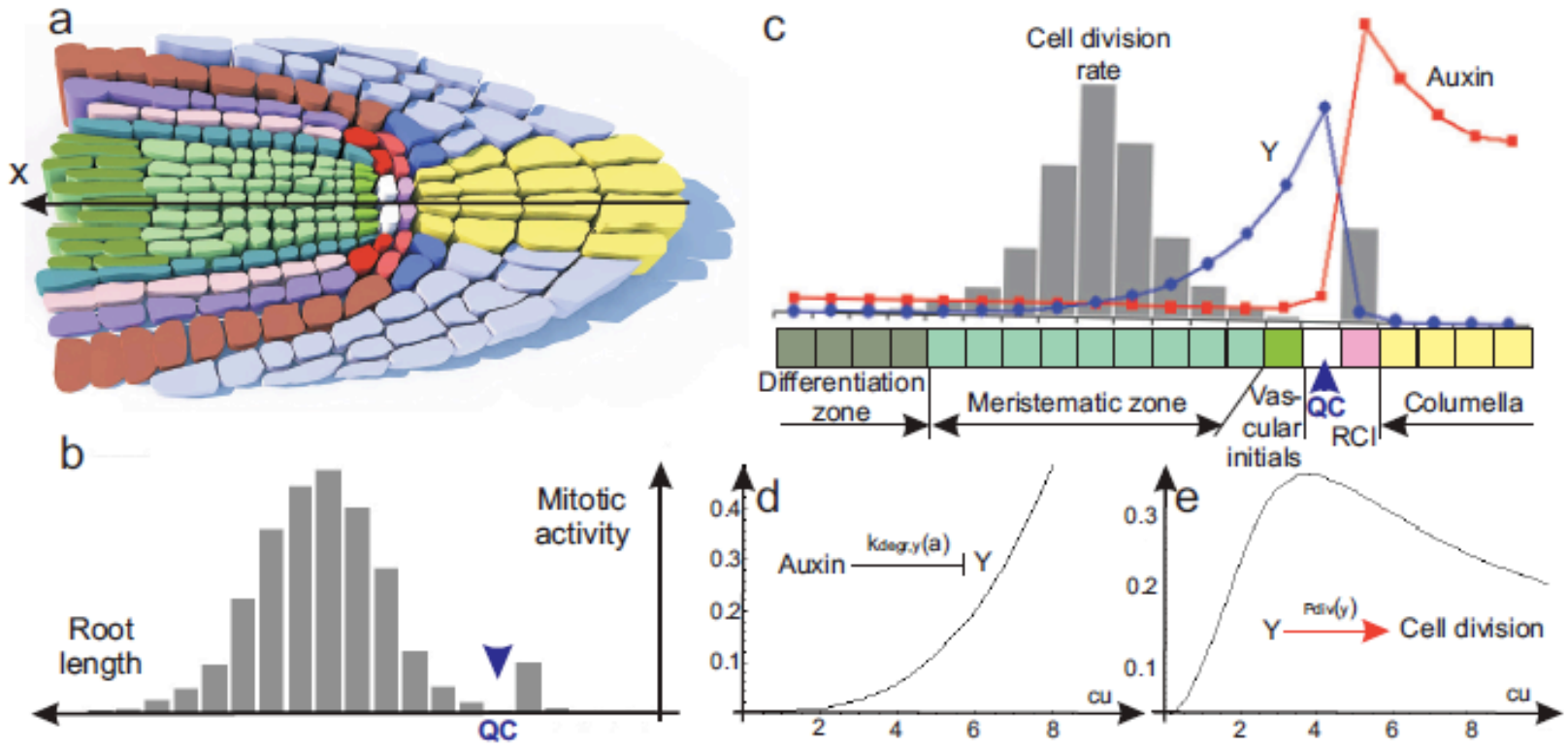
draw reaction r from distribution $k^{(r)}(x_{\text{in}}) / k^{(\text{total})}$;

draw x_{out} from $p^{(r)}(x_{\text{out}} \mid x_{\text{in}})$ and *execute* reaction r;

} **until** $t \geqslant t_{\max}$

Phys. Biol. **10** (2013) 035009

Root apical meristem (RAM): Growth model



[Mironova et al., BMC Systems Biology 2010]

Dynamical Grammar for Root

Auxin diffusion

Active transport

$$\{c_i = Cell(x_i, r_i, m_i, a_i, y_i),\quad c_{i+1} = Cell(x_{i+1}, r_{i+1}, m_{i+1}, a_{i+1}, y_{i+1}),\quad s_{i,i+1} = spring(c_i, c_{i+1})\} \rightarrow \{c_i, c_{i+1}, s_{i,i+1}\}$$

$$\text{solving}\left\{\frac{da_{i+1}}{dt} = -K_0 a_{i+1} b(a_{i+1}), \frac{da_i}{dt} = K_0 a_{i+1} b(a_{i+1})\right\}$$

Auxin flow from the shoot

$$\{c_N = Cell(x_N, r_N, m_N, a_N, y_N)\} \rightarrow \{c_N\}$$

$$\text{solving}\left\{\frac{da_N}{dt} = \alpha_{init} + \frac{0.17t}{CellCycleTime}\right\}$$

Hypothetical substance Y

$$\{c_i = Cell(x_i, r_i, m_i, a_i, y_i)\} \rightarrow \{c_i\}$$

$$\text{solving}\left\{\frac{dy_i}{dt} = -y_i(K_{d,y}(a_i) + \frac{v(r_i)}{r_i}), \frac{dr_i}{dt} = v(r_i)\right\}$$

$$K_{d,y}(a_i) = k^0_{d,y}\left(1 + \left(\frac{a_i}{k^1_{d,y}}\right)^{h_{y,1}} \Big/ \left(1 + \left(\frac{a_i}{k^2_{d,y}}\right)^{h_{y,2}}\right)\right)$$

Cell division

$$\{Cell(x_i, r_i, m_i = 2, a_i, y_i)\} \rightarrow \left\{\begin{array}{l} Cell(x_l, \frac{r_l}{2}, m_l = 1, a_l, y_l),\ Cell(x_{l+1}, \frac{r_{l+1}}{2}, m_{l+1} = 1, a_{l+1}, y_{l+1}) \\ s_{l,l+1} = spring(c_l, c_{l+1})\} \rightarrow \{c_l, c_{l+1}, s_{l,l+1}\} \end{array}\right\}$$

Cell growth

$$\text{with}\ \rho_{div}(y_i) = \left(\frac{y_i}{k_{div,1}}\right)^{h_{div,1}} \Big/ \left(1 + \left(\frac{y_i}{k_{div,2}}\right)^{h_{div,2}}\right)$$

[Mironova et al., BMC Systems Biology 2010]

Declarative Root growth model in Plenum

```
gRootGrowth := Grammar[rules →
    {
     (*continuous change in cell c1 radius *)
     {c1 = cell[cellID1, 1(*growth mode*),
         loc1, rad1, auxin1, y1, cellIDP, cellIDN]} → c1,
     solving[rad1' == cellGrowthLocFunc[rad1]],

     (*continuous change in cell c1 location *)
     {c1 = cell[cellID, cMode, loc, rad, auxin, y, cellIDPrev, cellIDNext], c2 =
        cell[cellIDNext, cModeN, locN, radN, auxinN, yN, cellID, cellIDNN]} → {c1, c2},
     solving[loc' == gGrowthModelMult * springXFunc[loc, rad, locN, radN]],


     (*continuous change in cell c1 location *)
     {c1 = cell[cellID, cMode, loc, rad, auxin, y, cellIDPrev, cellIDNext], c2 =
        cell[cellIDPrev, cModeP, locP, radP, auxinP, yP, cellIDPP, cellID]} → {c1, c2},
     solving[loc' == gGrowthModelMult * springXFunc[loc, rad, locP, radP]],

     (*change cell mode from growth to wait,when over a radius threshold *)
     cell[cellID, 1, loc, rad, auxin, y, cellIDPrev, cellIDNext] →
      cell[cellID, 2, loc, rad, auxin, y, cellIDPrev, cellIDNext],
     with[gGrowthModelMult * stopGrowthConst *
       grammarSigmoid[rad - gLimitCellRad, gDivideTemp]],

     ...
```

Modeling language intertranslation: "Cambium" flexible arrows

- Arrow[*type*[attributes], LHS, MS, RHS, params]
  - Eg: Arrow[Cellzilla[det, massaction, reversible, external], $\{R[i], L[j]\}_*, \varnothing, \{(R \diamond L)[i]\}_*$, {With[$k_f, k_r$], Using[nbr, i, j, Cell]}]
- Semantics defined by *operator* or by *reduction*
- Define & claim your own rule namespace
  - DynGram, LSystem, SBML, Cellerator, …
    - attributes: stoch/det, reversible, ratelaw, ODE, fire-once, fast[α], composite, …
  - Intertranslation by metarule
- Preliminary Cambium/Plenum and Cambium/Organism translators implemented in Mathematica
  - http://emj.ics.uci.edu/software/cambium-model-translation-software/

Variable-binding is *emergent* from Operator Algebra + integration

- Parameterized grammar rule operators:

$$\hat{O}_r = \int_{D_{\beta(1)}} \cdots \int_{D_{\beta(c)}} \cdots \left(\prod_c d\mu_{\beta(c)}(X_c) \right) \rho_r \left([x_i(\{X_c\})], [y_j(\{X_c\})] \right)$$
$$\times \left[\prod_{i \in \mathrm{rhs}(r)} \hat{a}_{a(i)}(x_i(\{X_c\})) \right] \left[\prod_{j \in \mathrm{lhs}(r)} a_{b(j)}(y_j(\{X_c\})) \right] \quad (19)$$

Thus, syntactic variable-binding has the semantics of multiple integration. This is the same result one would get if each rule with variables were replaced with a (finite, countable, or uncountably infinite) set of rules with all possible values substituted in for all the variables, with firing rates weighted by the relevant measure, and running in parallel.

[Annals of Math. and A. I., 47(3-4), January 2007]

- So, object parameters need measure spaces

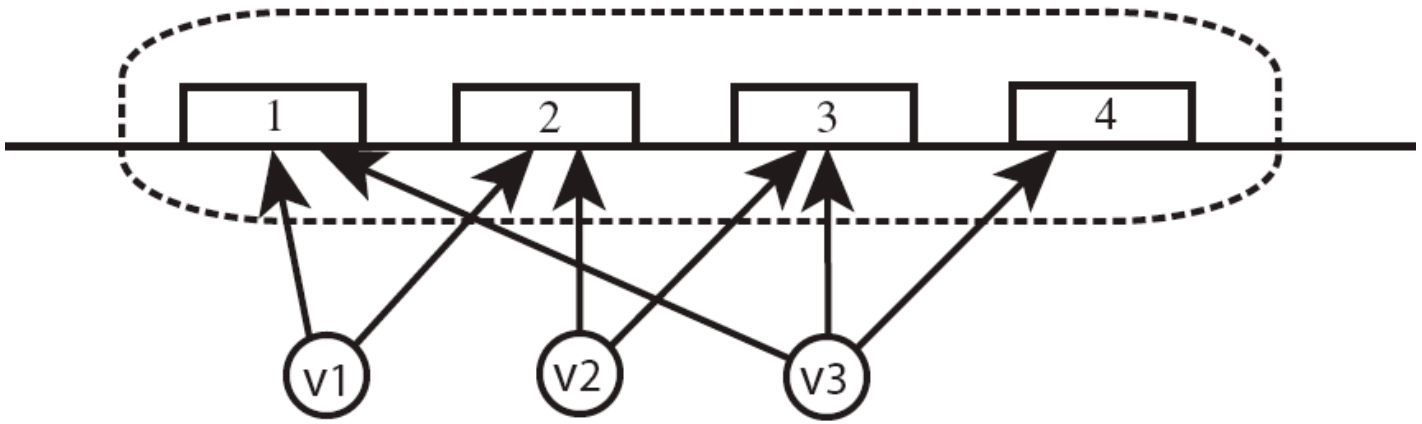
Expand on Underlying Biophysics

- Gene regulation networks
  - (transcription complexes)
- Signal transduction complexes
- Cytoskeleton

GRN ANN Equations

Model statement and its derivation from stat mech:

[Mjolsness Sharp and Reinitz, J. Theor. Biol. 152: 429-453, 1991]



$$\tau_i \dot{v}_i = g\left(\sum_j T_{ij} v_j + h_i\right) - \lambda_i v_i$$

Drosophila DV [~=Reeves et al.] in Cellzilla

Wolfram Mathematica | STUDENT EDITION Demonstrations | MathWorld | Wolfram Community | Help

```
In[126]:= ClearAll["Global`*"]

In[127]:= Needs["Cellzilla2D`"];
          << xlr8r.m;

In[139]:= cellRing = TemplateRing[72, 10, 1];
          n = NTissueCells[cellRing];

In[141]:= ShowTissue[cellRing, "CellNumbers" → True];

In[142]:= reactions = {
            {DOR ↦ SNA, GRN[1000, snaWidth, 1, snaθ]},
            {DOR ↦ ZEN, GRN[200, zenWidth, 1, zenθ]},
            {SNA + DOR ↦ VND, GRN[1, vndWidth, 2, vndθ]},
            {DOR + SNA ↦ SOG, GRN[100, sogWidth, 2, sogθ]},
            {SNA → ∅, snaDecay},
            {VND → ∅, vndDecay},
            {SOG → ∅, sogDecay},
            {ZEN → ∅, zenDecay}
          };
          intrpd = interpret[reactions]
```

Out[143]=

$$\left\{\left\{\text{DOR}'[t] == 0,\ \text{SNA}'[t] == \frac{1000}{1+e^{-\text{snaθ}-\text{snaWidth DOR}[t]}} - \text{snaDecay SNA}[t],\right.\right.$$

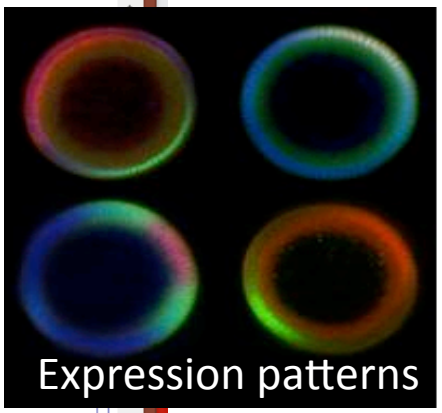
$$\text{SOG}'[t] == \frac{100}{1+e^{-\text{sogθ}-\text{sogWidth DOR}[t]^2-\text{sogWidth SNA}[t]^2}} - \text{sogDecay SOG}[t],$$

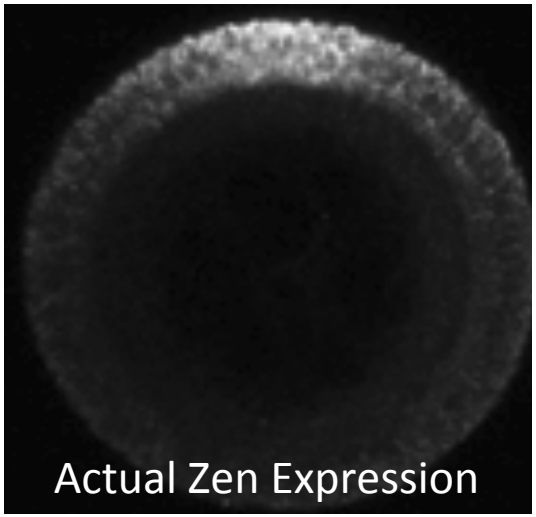
$$\text{VND}'[t] == \frac{1}{1+e^{-\text{vndθ}-\text{vndWidth DOR}[t]^2-\text{vndWidth SNA}[t]^2}} - \text{vndDecay VND}[t],$$

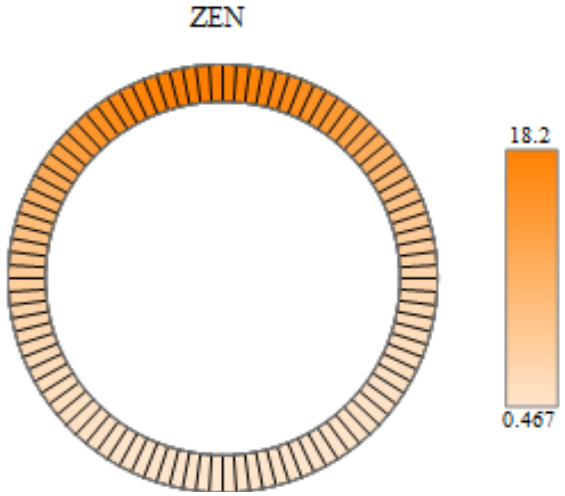
$$\left.\text{ZEN}'[t] == \frac{200}{1+e^{-\text{zenθ}-\text{zenWidth DOR}[t]}} - \text{zenDecay ZEN}[t]\right\},$$

$$\left.\{\text{DOR, SNA, SOG, VND, ZEN}\}\right\}$$

```
In[144]:= net = CelleratorNetwork[cellRing, "Reactions" → reactions,
              "Diffusion" → {{DOR, ddor}}];
```







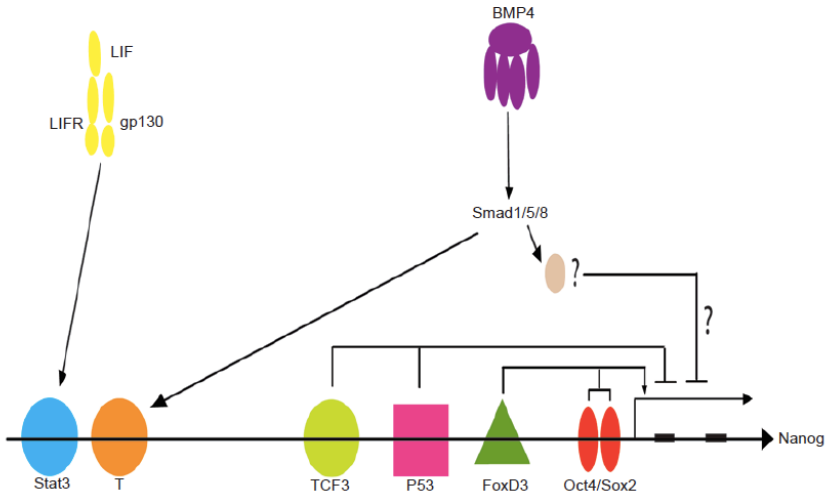
Kathryn Rodgers, UCI Computer Science PhD Student

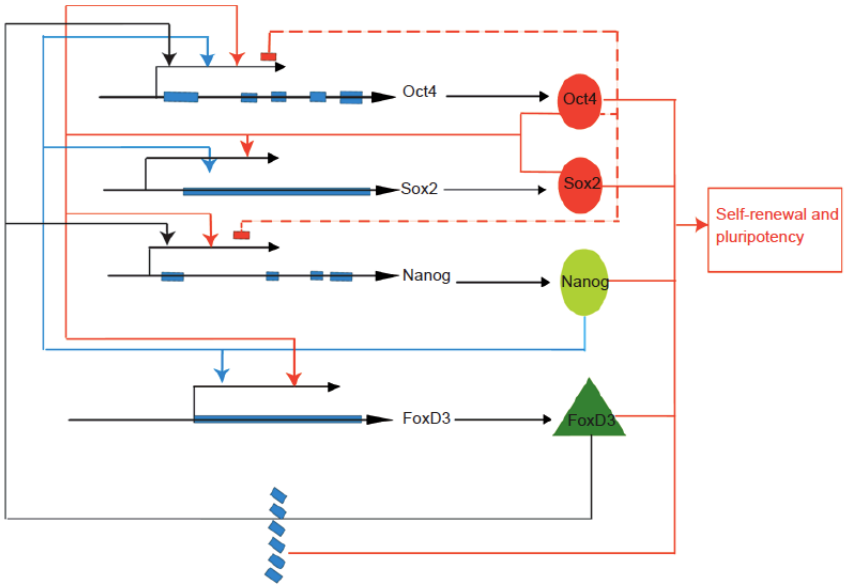
2x2 array of quasi-equilibrium cis-regulation models

	Pure stat mech (rational polynomial)	**ANN approximation**
Flat (1-level)	Proposition 1	MSR 1991 Proposition 2
CRM structure (2-level)	HCA 2002/HCA+ Proposition 3	Proposition 4

[Mjolsness J.Bioinformatics & Comp. Biology 5:2(b) 467-490, 2007]

ES cell switch





Pan and Thomson Cell Research (2007) 17: 42-49

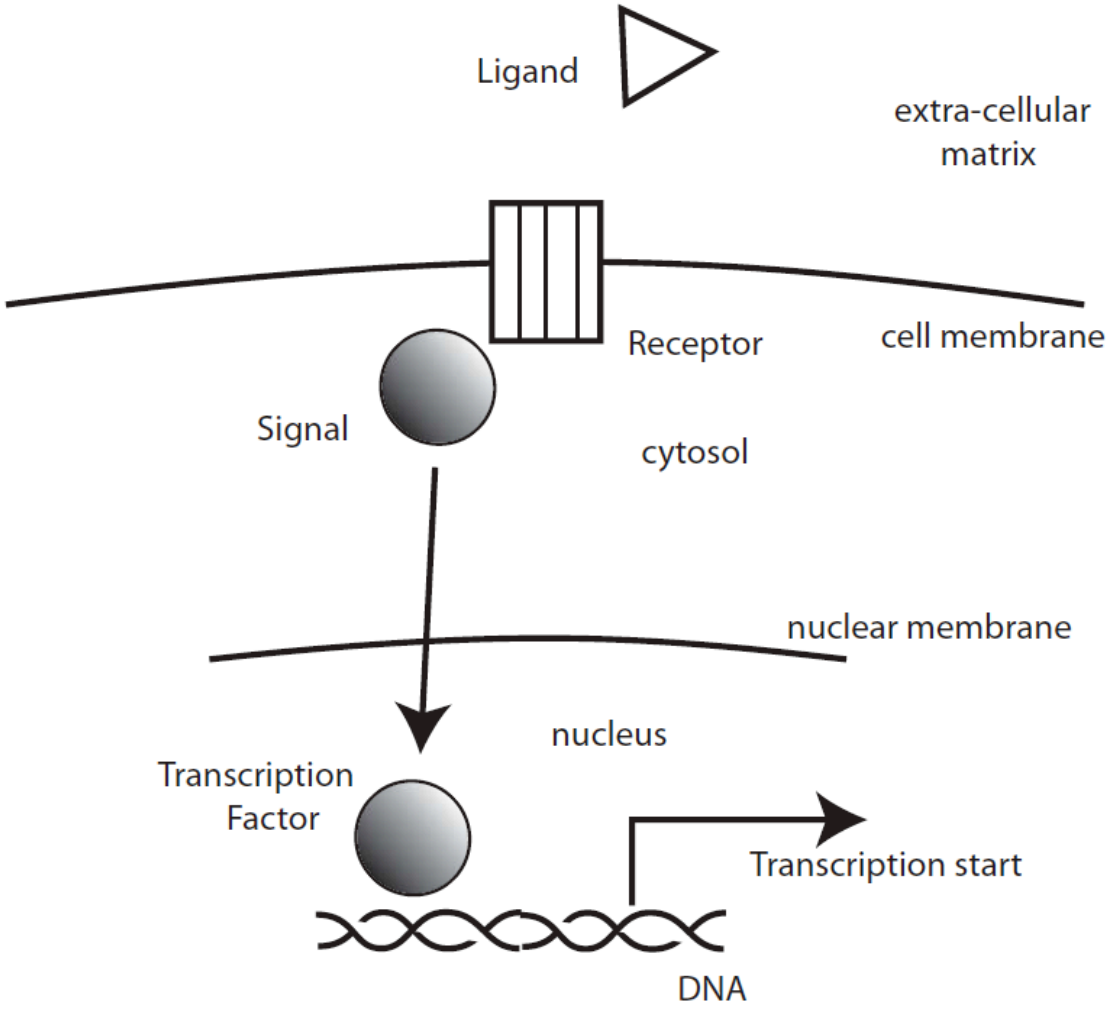
Chickarmane et al 2009:

$$\frac{d[O]}{dt} = \frac{a_0 + a_1[A] + a_2[O][S] + a_3[O][S][N]}{1 + b_0[A] + b_1[O] + b_2[O][S] + b_3[O][S][N] + b_4[C][O] + b_5[GC]} - \gamma_1[O] \quad (2)$$

$$\frac{d[S]}{dt} = \frac{c_0 + c_1[O][S] + c_2[O][S][N]}{1 + d_0[O] + d_1[O][S] + d_2[O][S][N]} - \gamma_2[S]$$

$$\frac{d[N]}{dt} = \frac{e_0 + e_1[O][S] + e_2[O][S][N]}{1 + f_0[O] + f_1[O][S] + f_2[O][S][N] + f_3[O][G]} - \gamma_3[N]$$

Signals to Genes



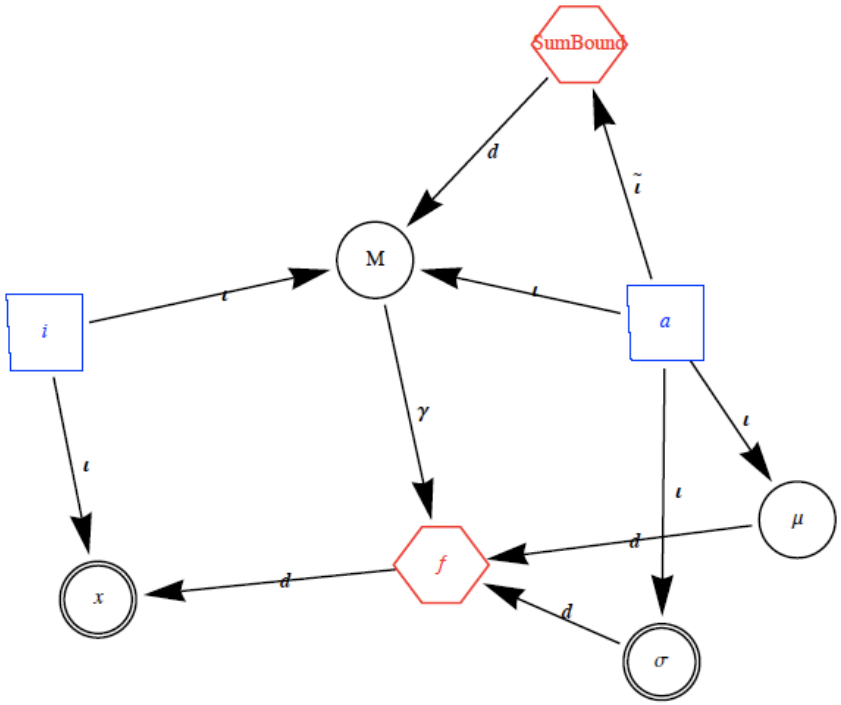
CaMKII example

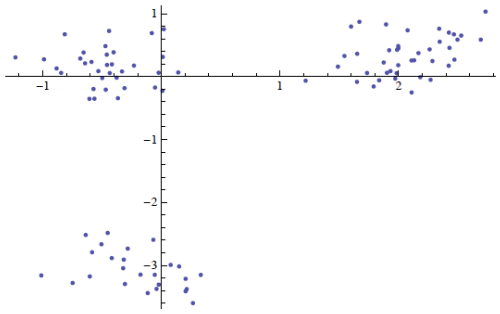
- Rule-based molecular complex modeling in Plenum / Dynamical grammars (excerpt):

```
(* CaM binding/unbinding free CaMKII *)
{CaM[n,c], CaMKII[num]} -> {Kk[n,c,0], CaMKII[num-1]},
  with[num*kon2[n,c,p0]/timeMultiplier],
{Kk[a0,b0,0], CaMKII[num]} -> {CaM[a0,b0], CaMKII[num+1]},
  with[koff2[a0,b0,0]If[a0>=0&&b0>=0,1,0]/timeMultiplier],

(* Dimerization *)
{Kk[a0,b0,p0], Kk[a1,b1,p1]} -> {Dimer[a0,b0,p0,a1,b1,p1]},
  with[If[p0<1||p1<1 && a0<=a1,kdimerize[a0,b0,p0,a1,b1,p1]/timeMultiplier,0]],
{Dimer[a0,b0,p0,a1,b1,p1]} -> {Kk[a0,b0,p0],Kk[a1,b1,p1]},
  with[kundimerize[a0,b0,p0,a1,b1,p1]/timeMultiplier],

(* phosphorylation *)
Dimer[a0,b0,p0,a1,b1,p1] -> {Kk[a0,b0,1],Kk[a1,b1,p1]},
  with[If[a0>=0&&b0>=0&&p0<1,kautotop[a0,b0,a1,b1,p1],0]/timeMultiplier],
Dimer[a0,b0,p0,a1,b1,p1] -> {Kk[a0,b0,p0],Kk[a1,b1,1]},
  with[If[a1>=0&&b1>=0&&p1<1,kautobot[a0,b0,p0,a1,b1],0]/timeMultiplier]
```

[Johnson PhD thesis 2012]. Original model: [Pepke et al. 2010]

Dependency Diagrams (DD):
Clustering Example



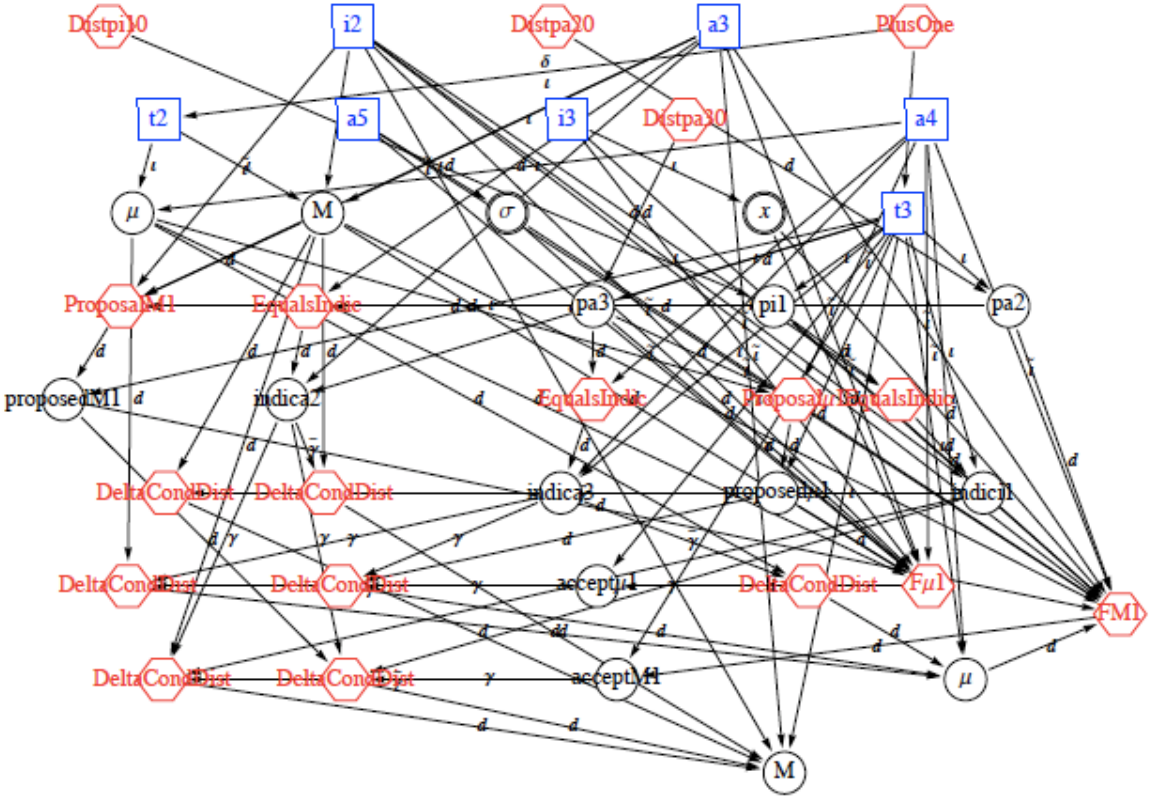


[Todd Johnson, UCI CS PhD thesis, 2012]

The semantics $\Psi(D)$ for this diagram approximates

$$\Pr(M,\mu|x,\sigma) = \frac{1}{Z}\left(\prod_{a=1}^{a_{\max}}\prod_{i=1}^{i_{\max}} N(x_i|\mu_a,\sigma_a)^{M_{ia}}\right)\left(\prod_{i=1}^{i_{\max}}\delta(\sum_{a=1}^{a_{\max}} M_{ia}-1)\right)$$

Autogenerated MCMC DD:
Clustering Example

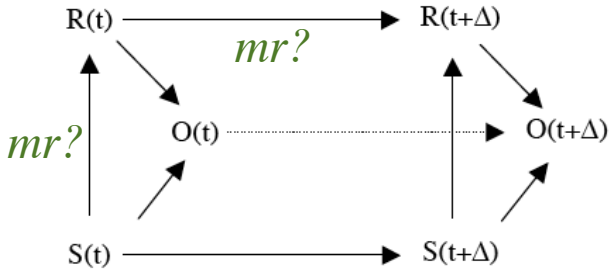


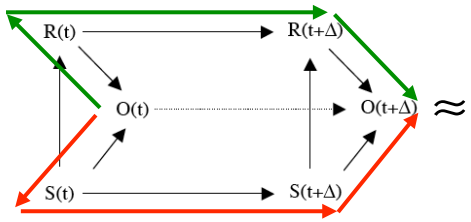
[Todd Johnson, UCI CS PhD thesis, 2012]

Figure 4.3: Automatically generated MCMC diagram for the clustering experiment. Several automatically generated functions and variables are present. DeltaCondDist

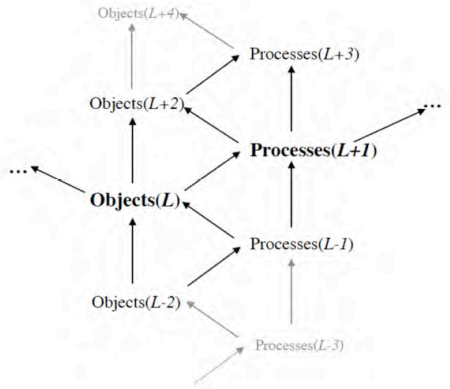
Model Reduction

- MR = commutative diagram + approximation + conditions of validity





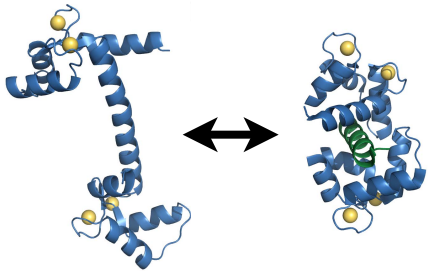
  - eg. from separation of scales in time or space
- stochastic S (or R) ↔ deterministic R (or S)
  - ⇒ *need both* for invariance of framework wrt MR



CaMKII example

- Rule-based molecular complex modeling in Plenum / Dynamical grammars (excerpt):

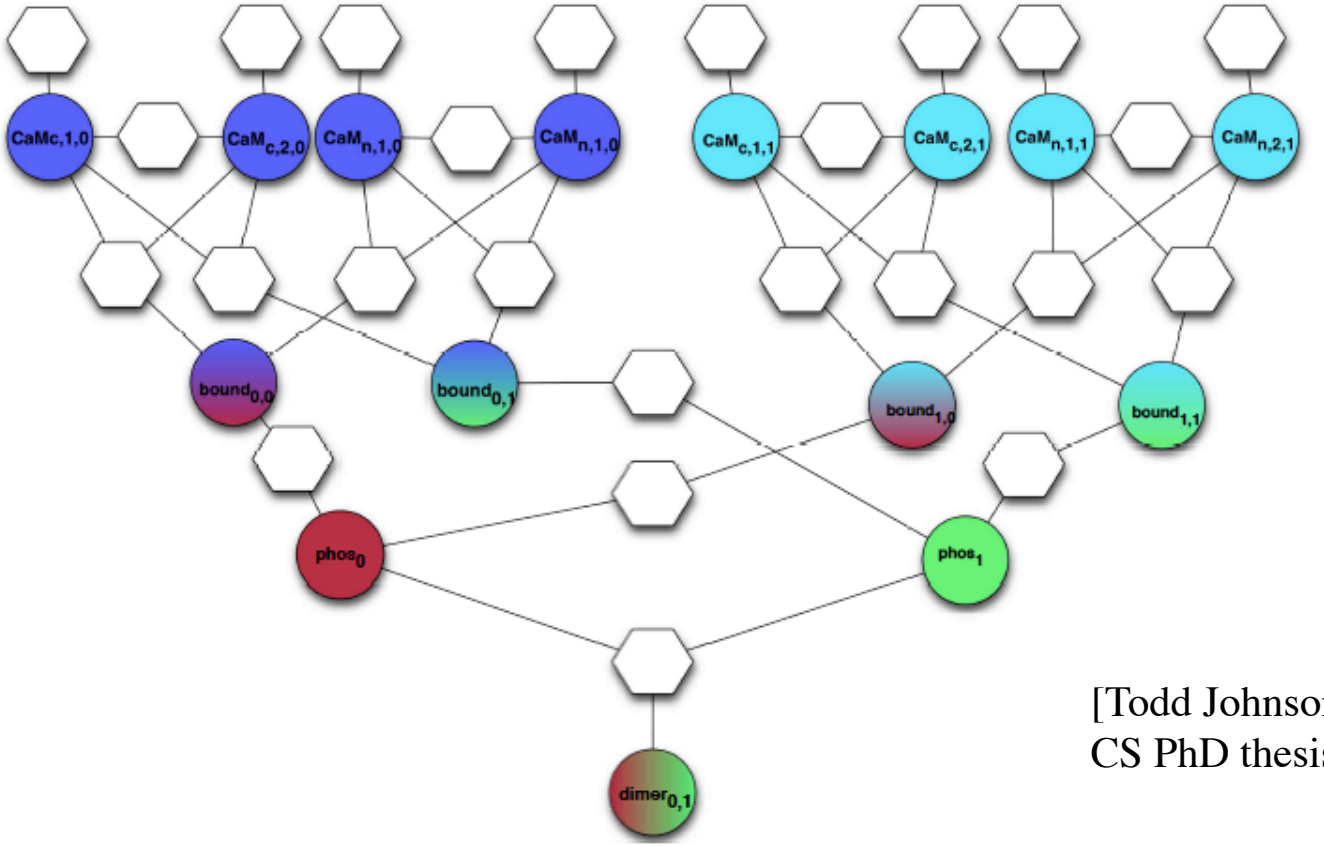
```
(* CaM binding/unbinding free CaMKII *)
{CaM[n,c], CaMKII[num]} -> {Kk[n,c,0], CaMKII[num-1]},
  with[num*kon2[n,c,p0]/timeMultiplier],
{Kk[a0,b0,0], CaMKII[num]} -> {CaM[a0,b0], CaMKII[num+1]},
  with[koff2[a0,b0,0]If[a0>=0&&b0>=0,1,0]/timeMultiplier],

(* Dimerization *)
{Kk[a0,b0,p0], Kk[a1,b1,p1]} -> {Dimer[a0,b0,p0,a1,b1,p1]},
  with[If[p0<1||p1<1 && a0<=a1,kdimerize[a0,b0,p0,a1,b1,p1]/timeMultiplier,0]],
{Dimer[a0,b0,p0,a1,b1,p1]} -> {Kk[a0,b0,p0],Kk[a1,b1,p1]},
  with[kundimerize[a0,b0,p0,a1,b1,p1]/timeMultiplier],

(* phosphorylation *)
Dimer[a0,b0,p0,a1,b1,p1] -> {Kk[a0,b0,1],Kk[a1,b1,p1]},
  with[If[a0>=0&&b0>=0&&p0<1,kautotop[a0,b0,a1,b1,p1],0]/timeMultiplier],
Dimer[a0,b0,p0,a1,b1,p1] -> {Kk[a0,b0,p0],Kk[a1,b1,1]},
  with[If[a1>=0&&b1>=0&&p1<1,kautobot[a0,b0,p0,a1,b1],0]/timeMultiplier]
```



[Pepke et al., PLoS Comp Bio, 2010]

[Johnson PhD thesis 2012]. Original model: [Pepke et al. 2010]

CaMKII Signaling Model: DD



[Todd Johnson, UCI CS PhD thesis, 2012]

Figure 7.1: An MRF model of calcium binding, CaM/CaMKII interaction, and CaMKII dimerization.

GCCD: Target and Approximate Stochastic Dynamics

- Target stoch. dynamics: Chemical master equation

$$\frac{dp}{dt} = W \cdot p \qquad i.e.\ \frac{d\ p([n_i])}{d\,t} \simeq \sum_r \rho^{(r)} \left(\prod_j (n_j - S_j^{(r)})_{\tilde{m}_j^{(r)}} \right) p([n_i - S_i^{(r)}]) - \sum_r \rho^{(r)} \left(\prod_j (n_j)_{\tilde{m}_j^{(r)}} \right) p([n_i])$$

- Approximation: Boltzmann/MRF + parameter ODEs

$$p(X) = q(X)/Z \qquad q(X) = \prod_c \phi_c(X_c) \qquad \phi_c(X_c) = e^{-\mu_c V_c(X_c)}$$

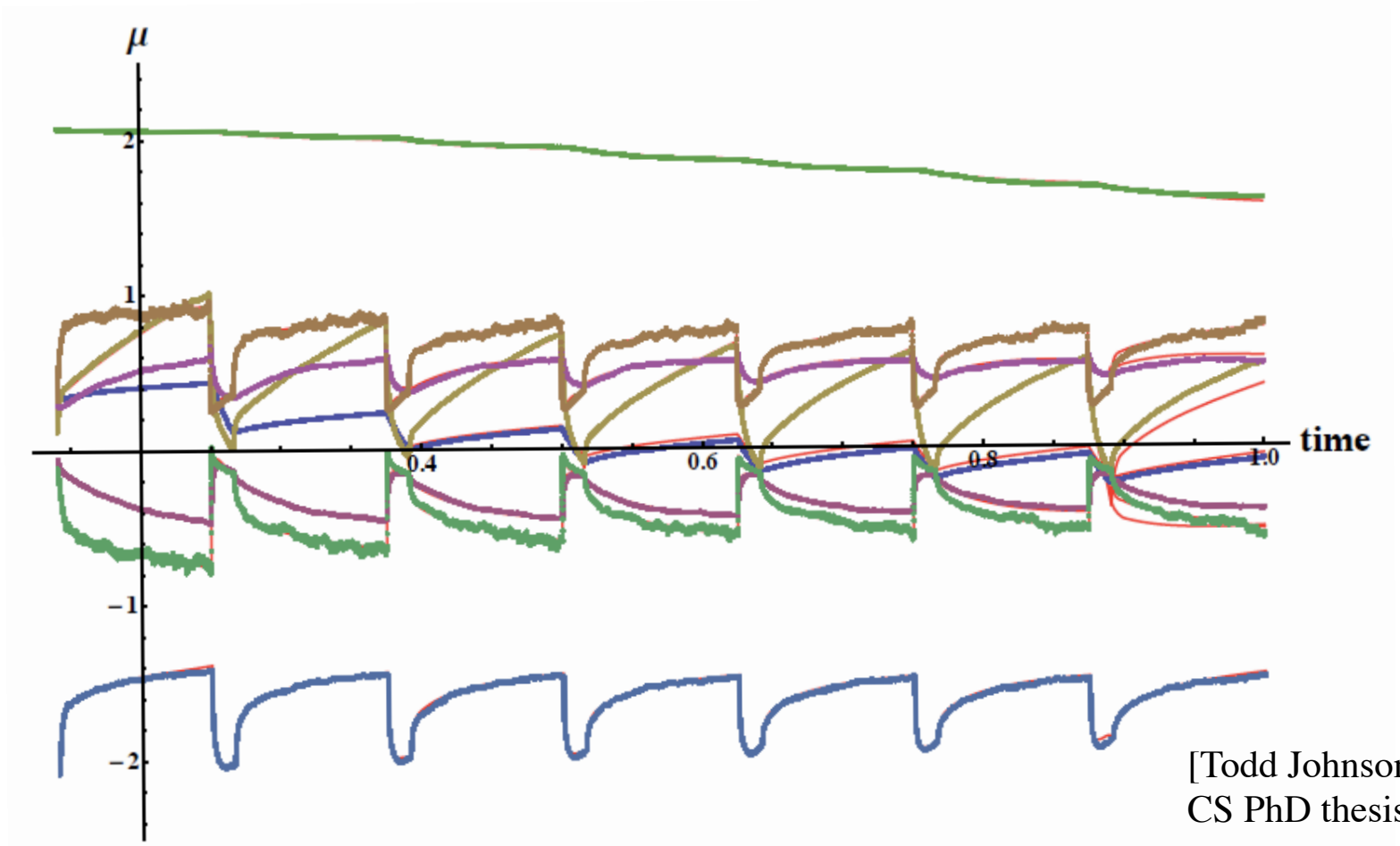
$$\frac{d}{dt}\mu_\alpha = f_\alpha(\mu|\theta) = \sum_A \theta_A f_{\alpha A}(\mu)$$

- Error criterion: KL Divergence

$$\mathcal{D}_{\mathcal{KL}}(\mu(t)) = -\int \tilde{p}(x;\mu(t)) \log\left(\frac{p(x;t)}{\tilde{p}(x;\mu(t))}\right) \mathrm{d}x$$

- Name: Graph-Constrained Correlation Dynamics
  - "Graph" = assumed MRF structure graph; "Correlations" = $\mu_c V_c(X_c)$

GCCD on CaMKII problem



[Todd Johnson, UCI CS PhD thesis, 2012]

Figure 7.12: Set of ordinary differential equations with learned coefficients (red lines) versus time series of eight MRF parameter values (colored lines) (MCell), spike train.

HiER-Leap scaling with reaction channels

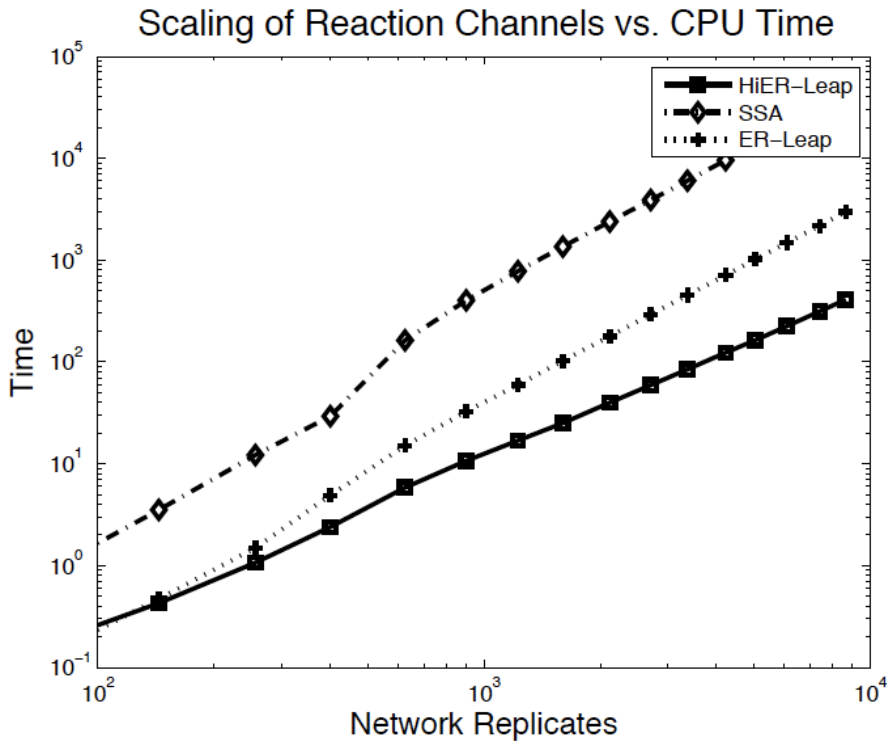


Figure 4.1: The Williamoski-Rossler model as seen in section 3.3.3 is used for this experiment. There are different number of network replicates on a 2D square grid with diffusion rate of 0.1. The number of replicates ranges from 4 . . . 8649 which equates to 64 . . . 189612 reaction channels.

HiER-Leap

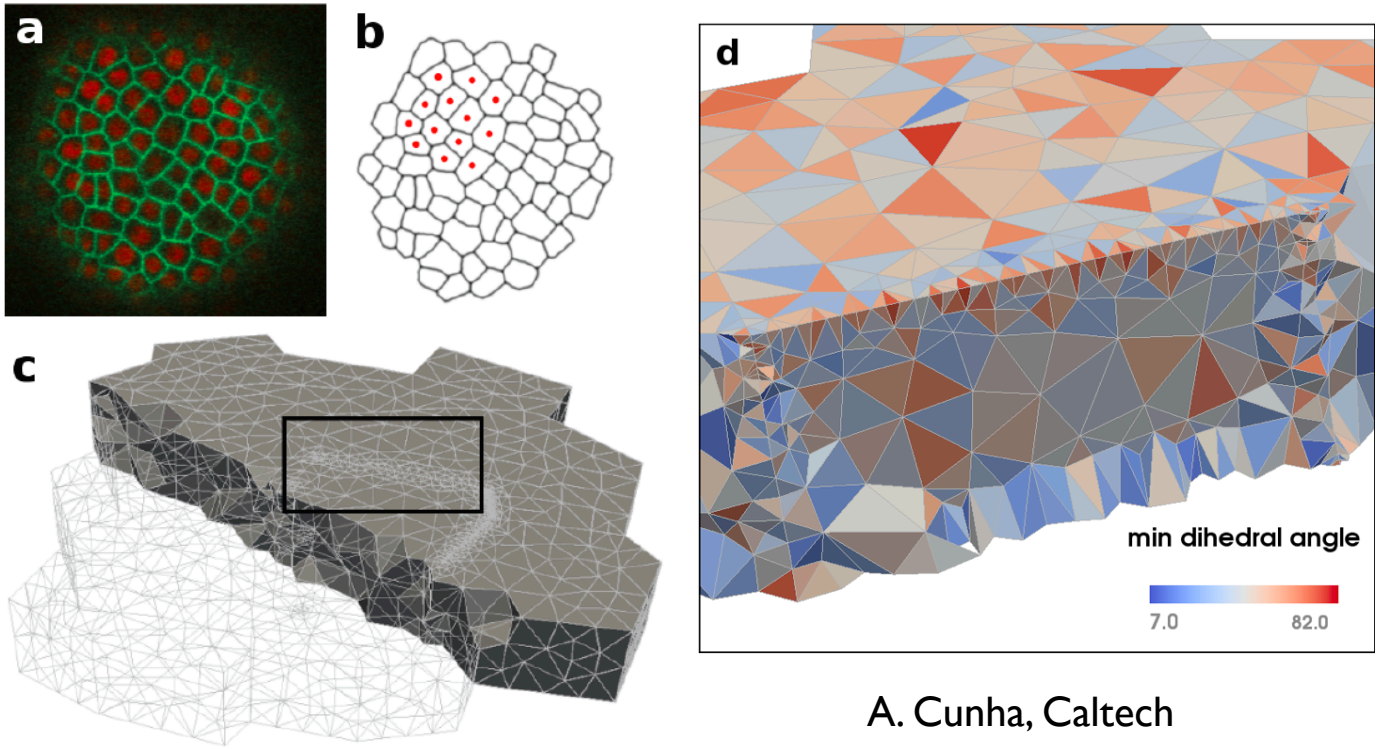
- Hierarchical version of ER-Leap
  - "blocks" of reaction channels
  - sparsely connected through shared reactants
  - similar to ER-Leap at coarse and fine scales
    - multinomial distribution over blocks, and again over reaction channels within each block
  - so far, only 2 levels tried out
- Reaction blocks permit better bounds
  - don't assume all reactant numbers have changed as much as possible
    - the block propensity changes are then too pessimistic by a factor of |block|
  - instead, assume the "worst" reaction in each block happened L times
    - => tighter bound on block propensities
- Parallelism is intrinsic - by reaction block
  - slightly coarser: allows multithread load balancing within processors

Extended Objects

- Graphs - already handled above
- Continuua - need something new

Implementations:

Finite Element Methods



A. Cunha, Caltech

Cell complexes are natural computational representations for many biological cells. (a) A horizontal slice of a three dimensional confocal microscope image of live plant cells (in a shoot apical meristem) which form a tightly packed connected network. These epidermal plant cells are naturally approximately convex-polyhedral; cells in other tissues can be decomposed into such shapes. Fluorescent green cell membrane marker highlights the cell walls; cell nuclei are marked with red. (Image courtesy of Marcus Heisler). (b) Automatic image segmentation followed by hand marking of a subset of 13 cells. (c) The marked cells are given a uniform thickness and then meshed with nonuniform sized tetrahedra useful for finite element simulations. The cutaway visualization displays an output mesh created using Tetgen [29]. Color coding of the quality of tetrahedra according to a minimum dihedral angle criterion.

Lower dimension strata => higher resolution

Escalation

Eg. ODE to PDE

Ordinary differential object	Partial differential object
d/dt	d/dt
$i \in \mathbb{N}$ (lattice index)	$x \in \mathbb{R}$ (coordinate)
x_i	$\Phi(x)$
y_i	$\Phi'(x)$
$\Delta_{F\|B\ i}$ (finite difference, forward or backward)	$\partial/\partial\ x_i$ (partial derivative)
$\partial/\partial\ x_i$ (partial derivative)	$\delta/\delta\,\Phi(x)$ (functional derivative)
D (homog. scalar diffusion coef.)	D (homog. scalar diffusion coef.)
$\delta(y-x) = \prod_i \delta(y_i - x_i)$	$\Delta(\Phi' - \Phi) = \prod_x\ \delta(\Phi'(x) - \Phi(x))$
$\int dx\, g(x)$ (ordinary integral)	$\int \mathcal{D}\Phi\, G[\Phi]$ (functional integral)
$a_\tau\ (x) = a_\tau([x_i])$	$a_\tau(\Phi) = a_\tau(x \mapsto \Phi(x))$

- ... Requires a function →, and limiting process on observables

Proposed syntax for PDEs

- Syntax:

$$S = \text{LevelStratum}(\ldots, x, \ldots, \varphi(x), \partial\text{parents}, \partial\text{children}) \longrightarrow S$$

$$\textbf{solving } \frac{d\,\varphi(x,t;w)}{d\,t} = F[\varphi](x)$$

$$\textbf{subject to } \text{FiniteSobolevNorm}[\varphi, p]$$

  - Boundary conditions: d-1 dimensional child strata

- Method:

$$f(x) = \sum_{s=0}^{\infty} \left(\sum_{m \in \text{ bases attached to cells at scale } s} c_{s\,n}\, b_{s\,n}(x, [\varphi_{s'} \mid s' < s]) \right)$$

- Semantics:

$$\hat{O}_{\text{drift}} = -\int\int \mathcal{D}\Phi\, \mathcal{D}\Phi'\ \hat{a}(\Phi')\, a_\tau(\Phi) \left(\int dx\, \frac{\delta}{\delta\,\Phi'(x)}\, F[\Phi'](x)\, \Delta(\Phi' - \Phi) \right)$$

Summary: Progress in Computer-Algebraic Biomodeling Languages

- Spatial biomodeling w. cell division
  - Cellzilla: spatial reaction network modeling + cell division
  - Plenum: Rule-based hybrid modeling with dynamical spaces via graph grammars
- Sci apps: plant dev, Gene Regulation Networks
- Algorithms: ODE/SSA/vbl-binding; HiERLeap, ...
- Dependency Diagrams & GCCD learning:
  - super "graphical models" for ML, optimization => model reduction
- Needed: spatial continuua, geometries & PDEs
- See: http://emj.ics.uci.edu/

Abstract

- Abstract:

Collaborative projects have resulted in several Mathematica-implemented modeling languages aimed at general-purpose biological modeling, which is a useful and topical but indefinitely expandable goal. We update previous work on reaction arrow translation (Cellerator) and on "dynamical grammars" (Plenum). Automatic learning of parameters has been developed for (a) stochastic reaction networks and for (b) model reduction of such networks for molecular complexes, using "dependency diagrams". Spatial modeling methods have developed in the form of adding some Plenum-like cell-level rules to Cellzilla (a Cellerator add-on package), parallelizable exact stochastic simulation algorithms not yet included in computer algebra software, and a current theoretical project on deriving partial differential equation methods that can be integrated with existing hybrid numerical techniques.

- Joint work with Bruce Shapiro, Todd Johnson, and David Orendorff