Bioinformatics for Gene Regulatory Networks: Analysis at Three Different Scales

Prof. Theodore J. Perkins

BINF 621

Nov 26, 2008

Outline

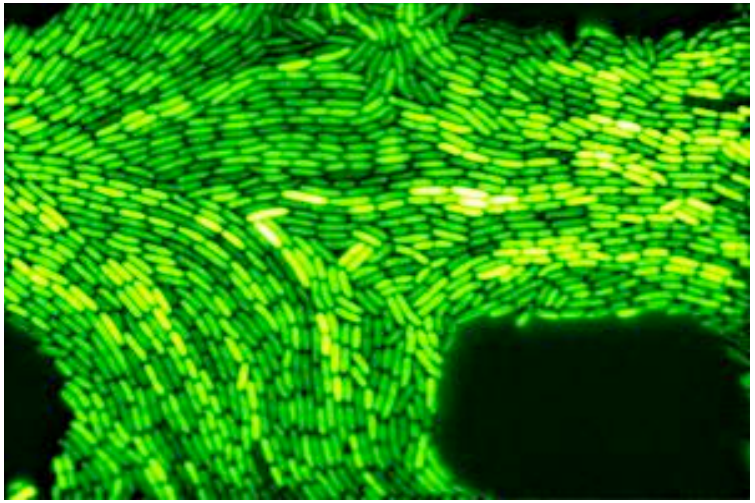
1. Estimating protein copy number from fluorescent imaging data
   Rosenfeld, Perkins, Alon, Elowitz, Swain, "A fluctuation method to quantify *in vivo* fluorescence data." *Biophysical Journal* (2006)

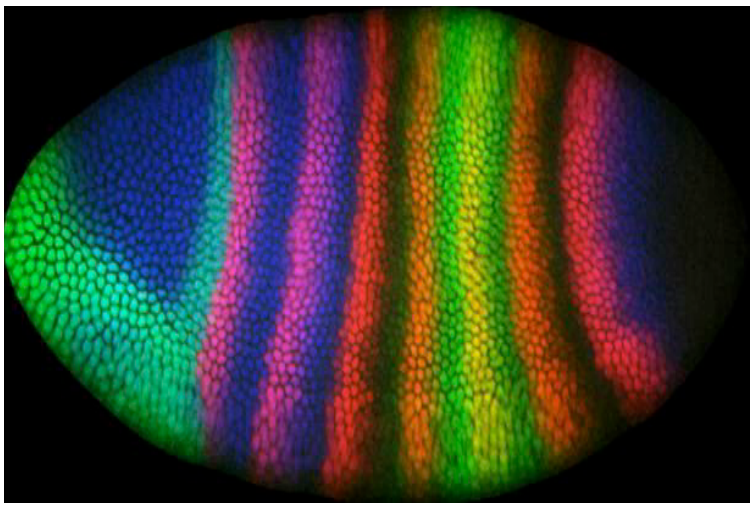
2. Modeling regulatory interactions in the gap gene system of *Drosophila melanogaster*
   Perkins, Jaeger, Reinitz, Glass, "Reverse Engineering the Gap Gene Network of Drosophila melanogaster." *PLoS Computational Biology* (2006)

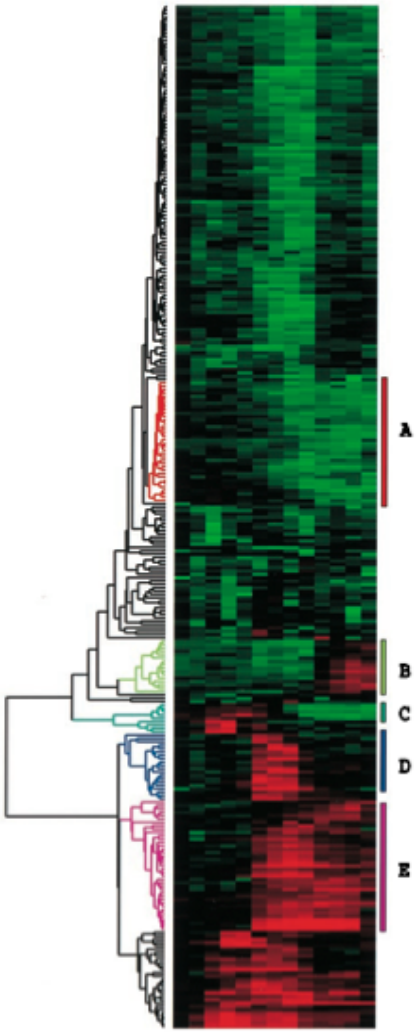
3. Searching for coherent subnetworks in large interaction networks
   Scott, Perkins, Bunnell, Pepin, Thomas, Hallett, "Identifying Regulatory Subnetworks for a Distinguished Set of Genes." *Molecular and Cellular Proteomics* (2005)

Part I: Estimating protein copy number from fluorescent imaging data

Imaging technologies give relative expression







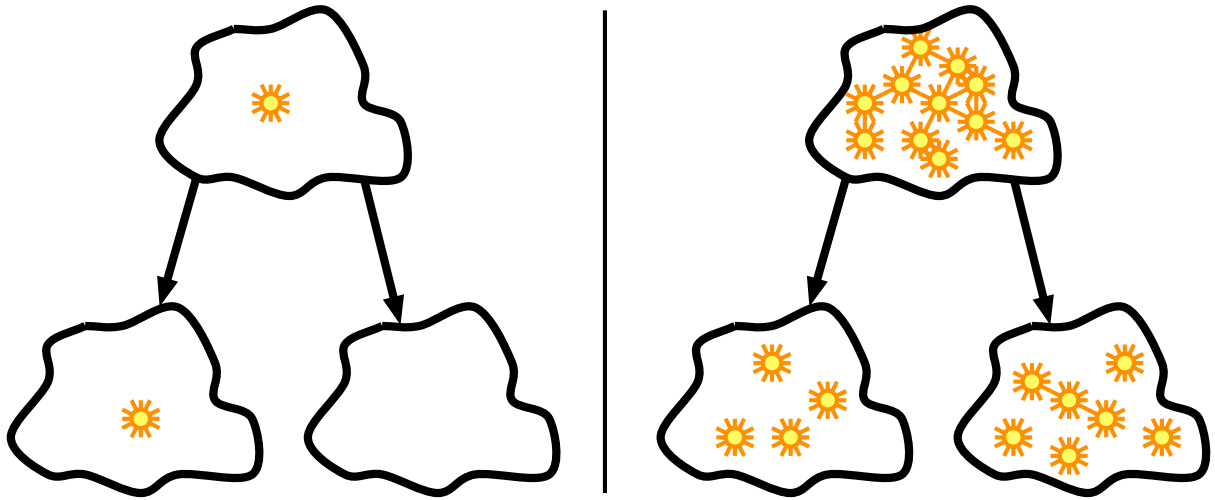
Relative expression versus copy number

- In many cases, relative expression is very useful
  Correlate or cluster expression of genes, look at effects of knock-outs & other interventions, infer function, infer co-regulation, look for disease markers, etc.

- But sometimes we want to know the copy number—the actual number of molecules present
  - To understand robustness to molecular noise
  - To choose appropriate modeling formalisms
  - To recreate realistic conditions *in vitro*
  - Estimate energetic costs

- In some cases, variability in fluorescent intensity can be used to estimate copy number

Intuition: Variability between daughter cells



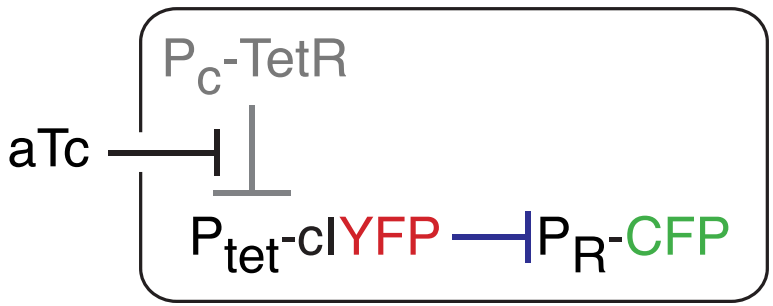
$\Rightarrow$ Difference in daughter cell intensities, as a fraction of parent, in relation to copy number

The experiment

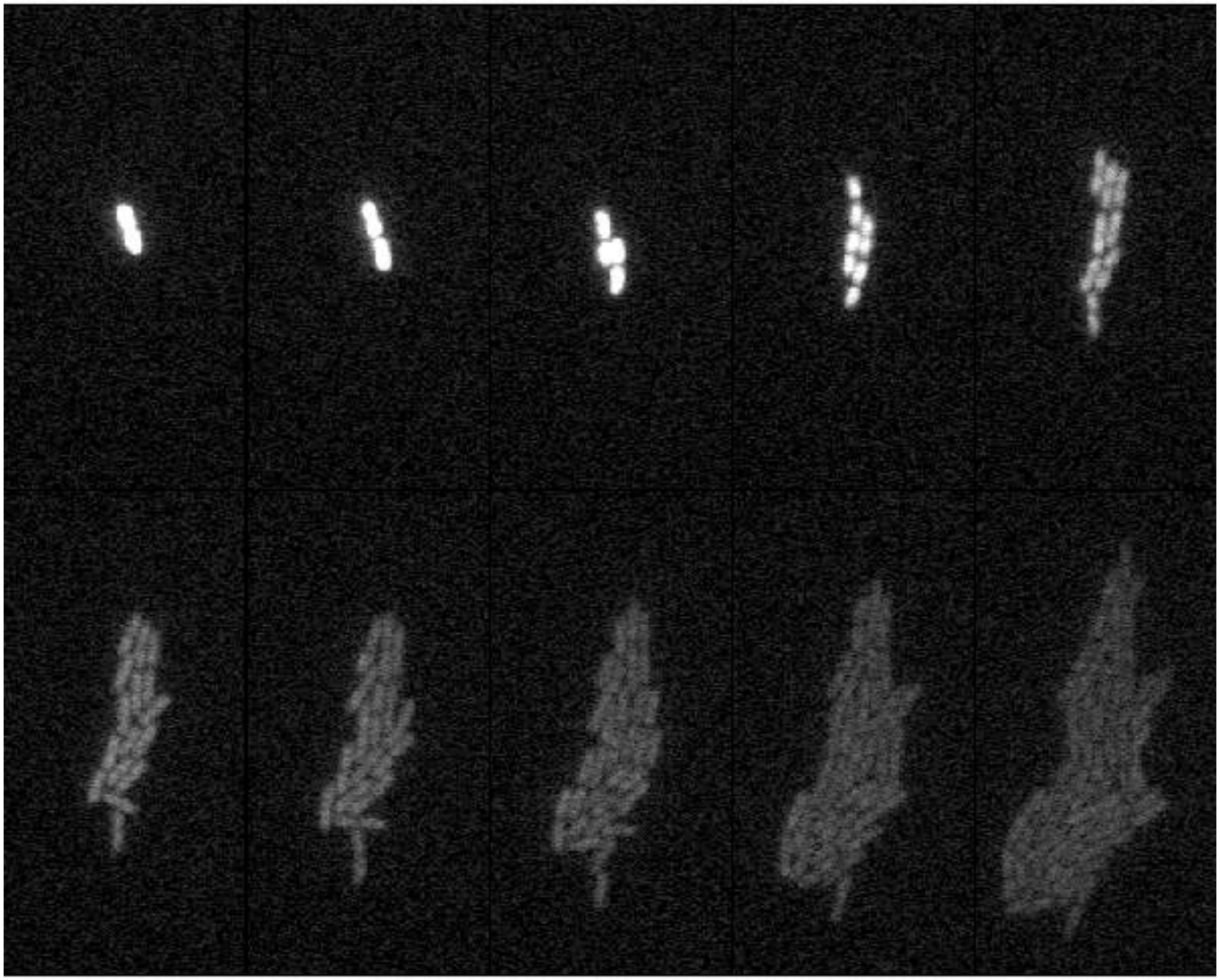
- Transcription of a fluorescent gene is turned on briefly then halted, resulting in a fixed, but unknown, number of fluorescent proteins.
- A series of fluorescent images capture relative expression levels as colony grows

Details:

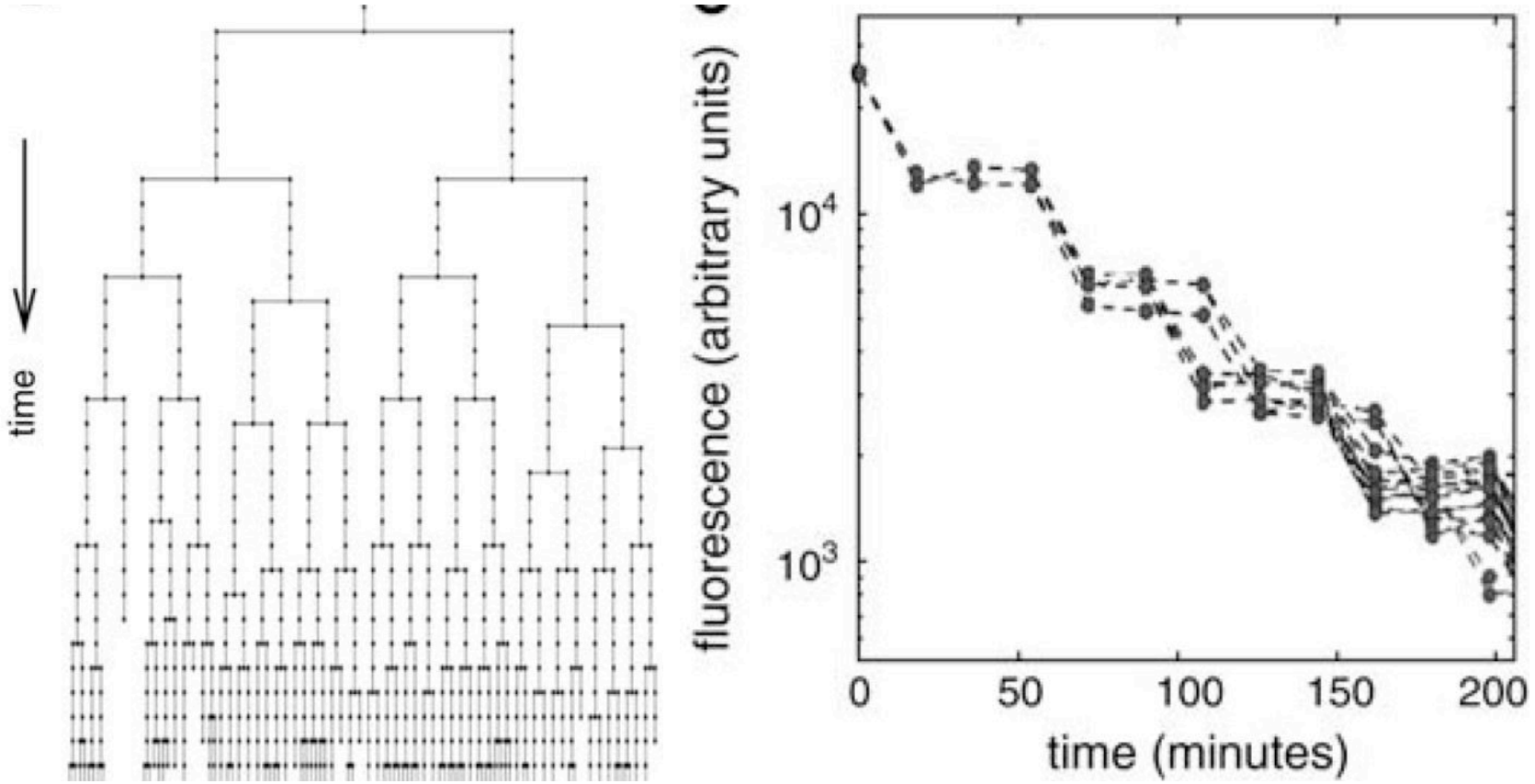
- The fluorescent protein is a fusion of λ-phage protein CI with YFP
- Transcription is repressed by ubiquitously expressed tetracycline repressor (TetR)
- Brief period of transcription achieved by spiking in anhydrotetracycline (aTc), which interferes with TetR, and then washing it out



The fluorescent image time series



Cell-by-cell "family tree" and intensities



$\Rightarrow$ Problem: Estimate protein copy number in each cell!

A simple model

We assume fluorescence is proportinal to protein copy number:

$$y_i = \nu n_i \ ,$$

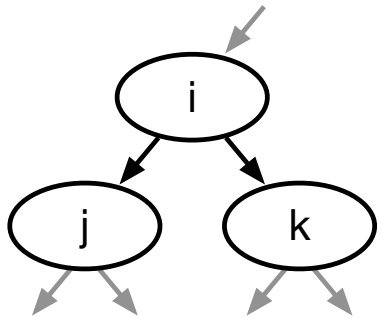
where

- y_i is observed fluorescence of cell i
- n_i is true protein copy number in cell i
- ν relates copy number to fluorescence

$\Rightarrow$ We will estimate ν, from which we can estimate the protein copy number in each cell ($n_i = y_i/\nu$).

A simple model (II)

Consider a single triad in the tree:

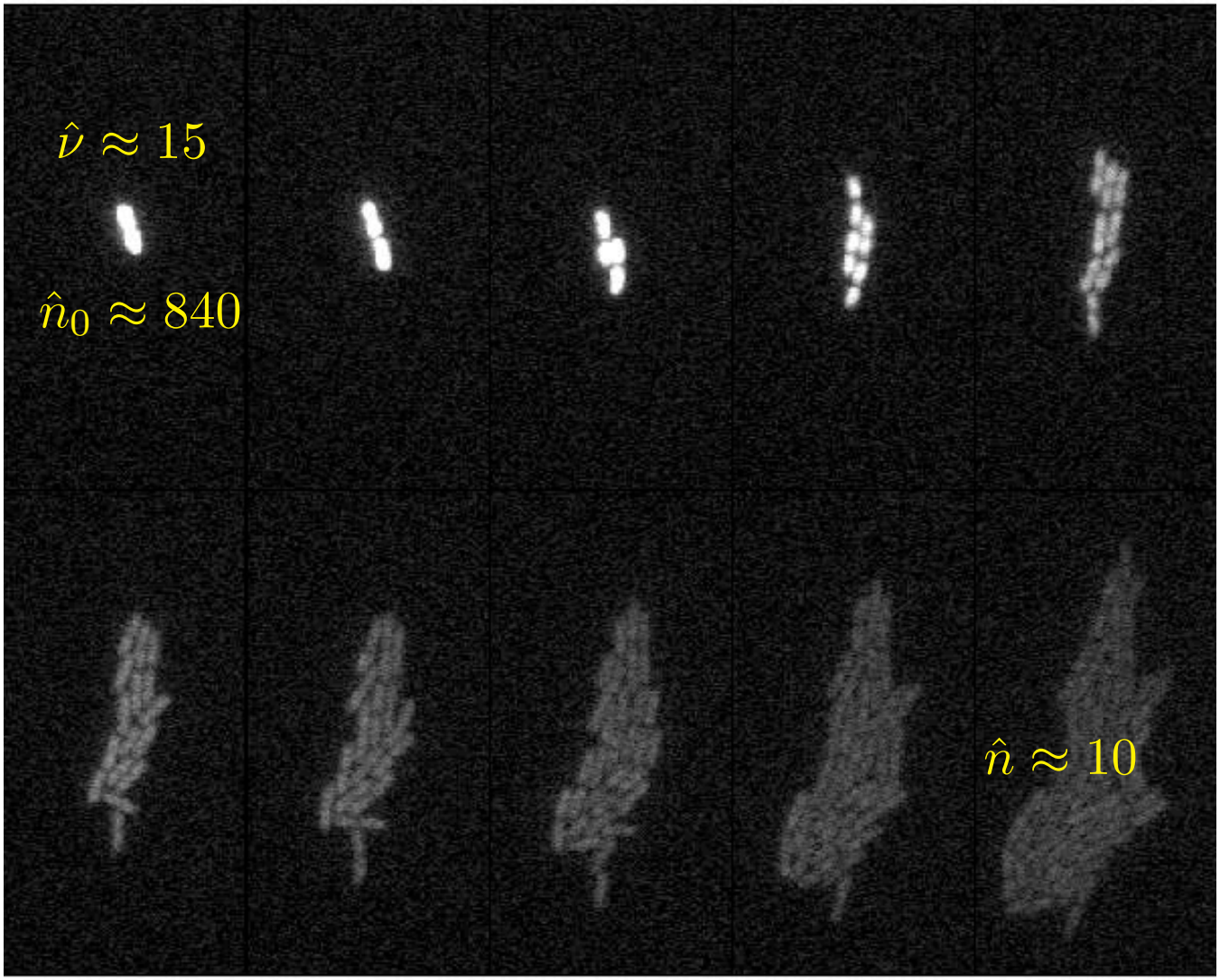


- Assume no protein is lost: $n_i = n_j + n_k$
- Assume $n_j, n_k \sim \text{Binomial}(n_i, \frac{1}{2})$
- Let:

$$\hat{\nu}_i = \frac{(y_j - y_k)^2}{y_i}$$

- Why? Because $E(\hat{\nu}_i | y_i) = \nu$.
- We estimate ν by averaging over all triads: $\hat{\nu} = \frac{1}{N} \sum_{i=1}^{N} \hat{\nu}_i$

Results of simple model



Accounting for noise

We assume:

- Additive Gaussian observation noise: $y_i = \nu n_i + \epsilon_i$, where $\epsilon_i = N(0, \sigma)$
- Binomial copy number inheritance: $n_{2i} \sim \text{Binomial}(n_i, \frac{1}{2})$
- Conservation of protein: $n_{2i+1} = n_i - n_{2i}$

Compute:

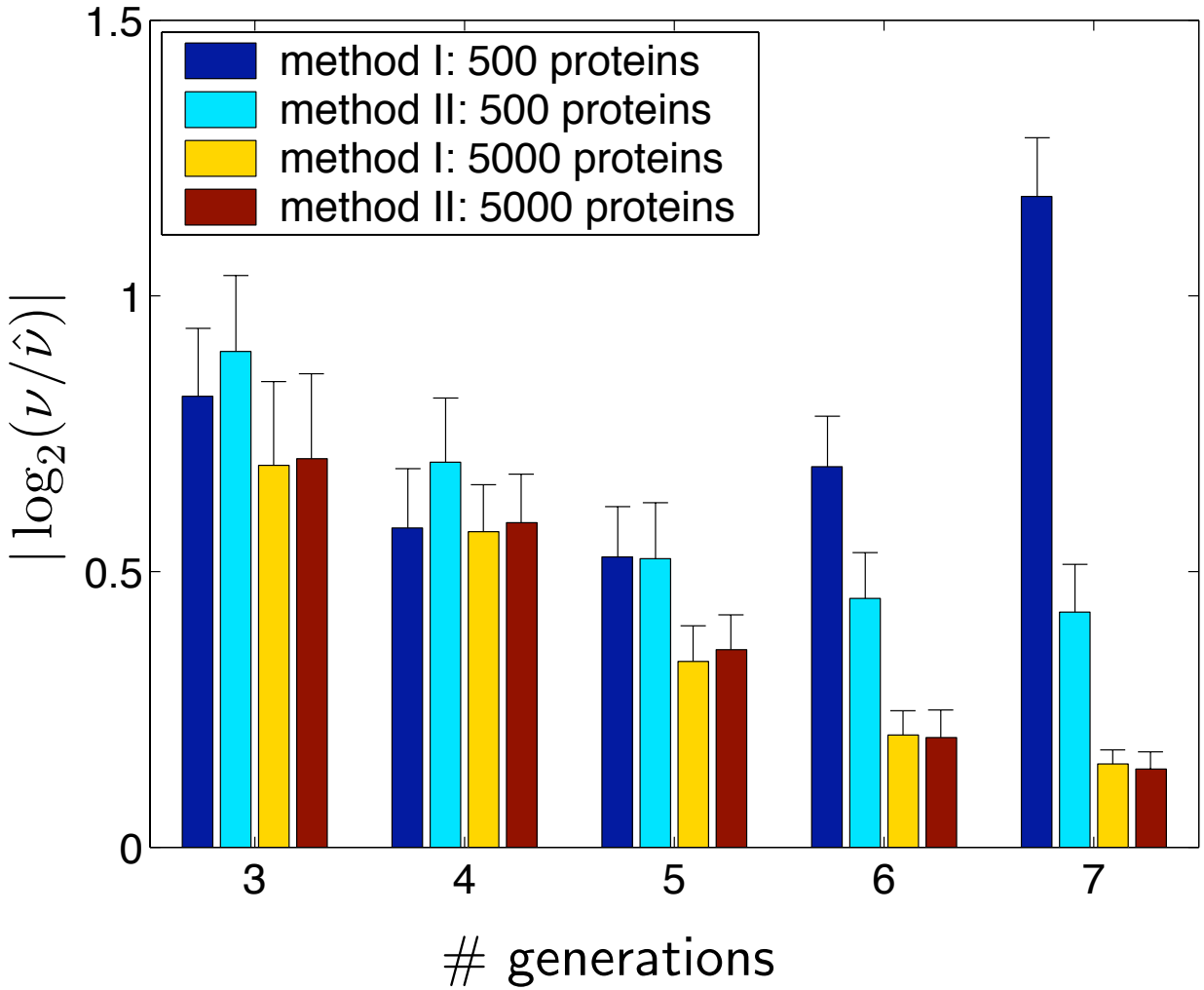
$$P(\nu, \sigma | y_1, \ldots, y_N) = \frac{P(y_1, \ldots, y_N | \nu, \sigma) P(\nu, \sigma)}{P(y_1, \ldots, y_N)} \ ,$$

where

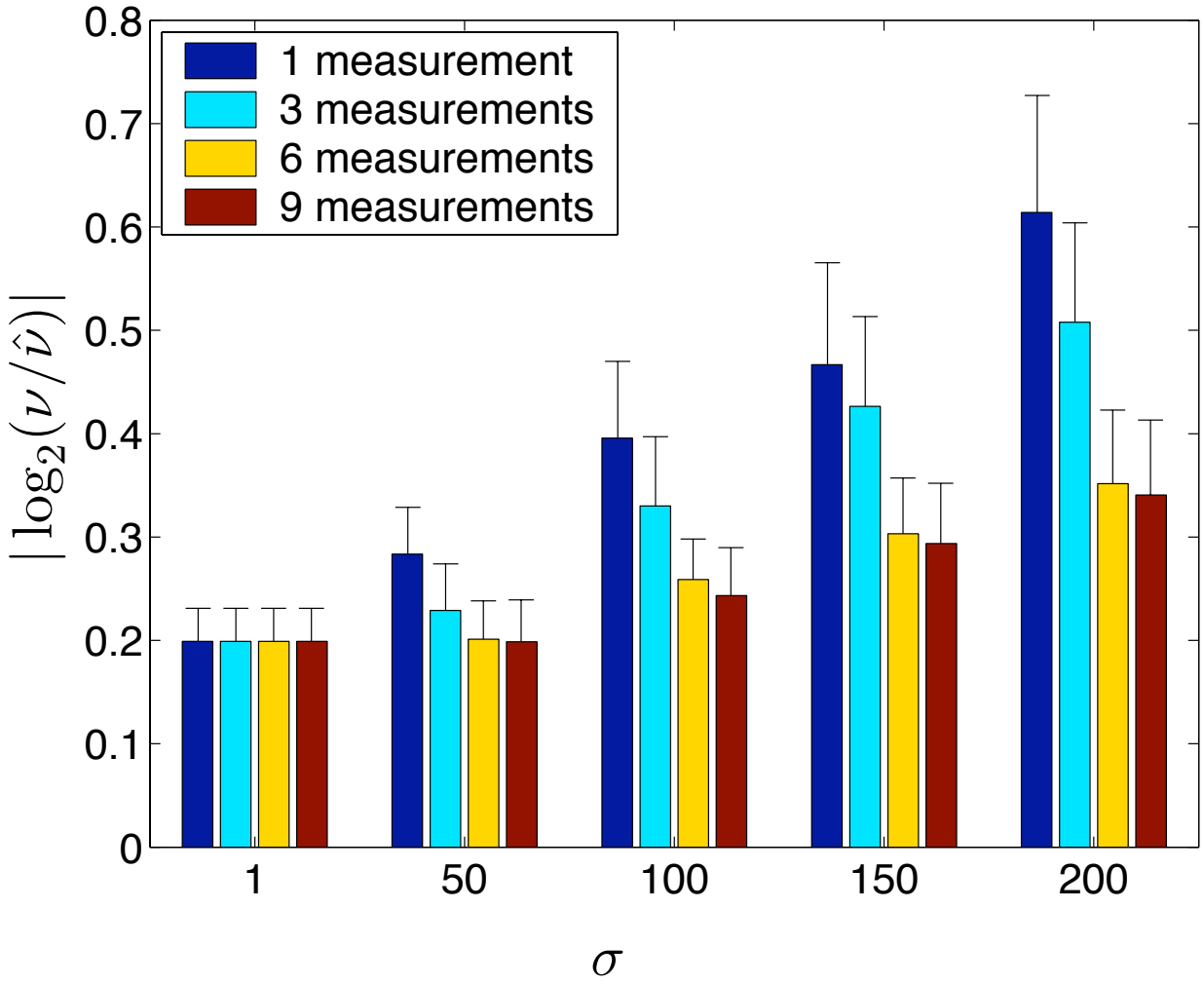
$$P(y_1, \ldots, y_N | \nu, \sigma) = \sum_{n_1, \ldots, n_N} \frac{P(y_1, \ldots, y_N | n_1, \ldots, n_N, \nu, \sigma) \times}{P(n_1, \ldots, n_n)}$$

Naive computation of summation infeasible, but can be transformed to series of N 1-dimensional summations (similar to Felsenstein's algorithm).

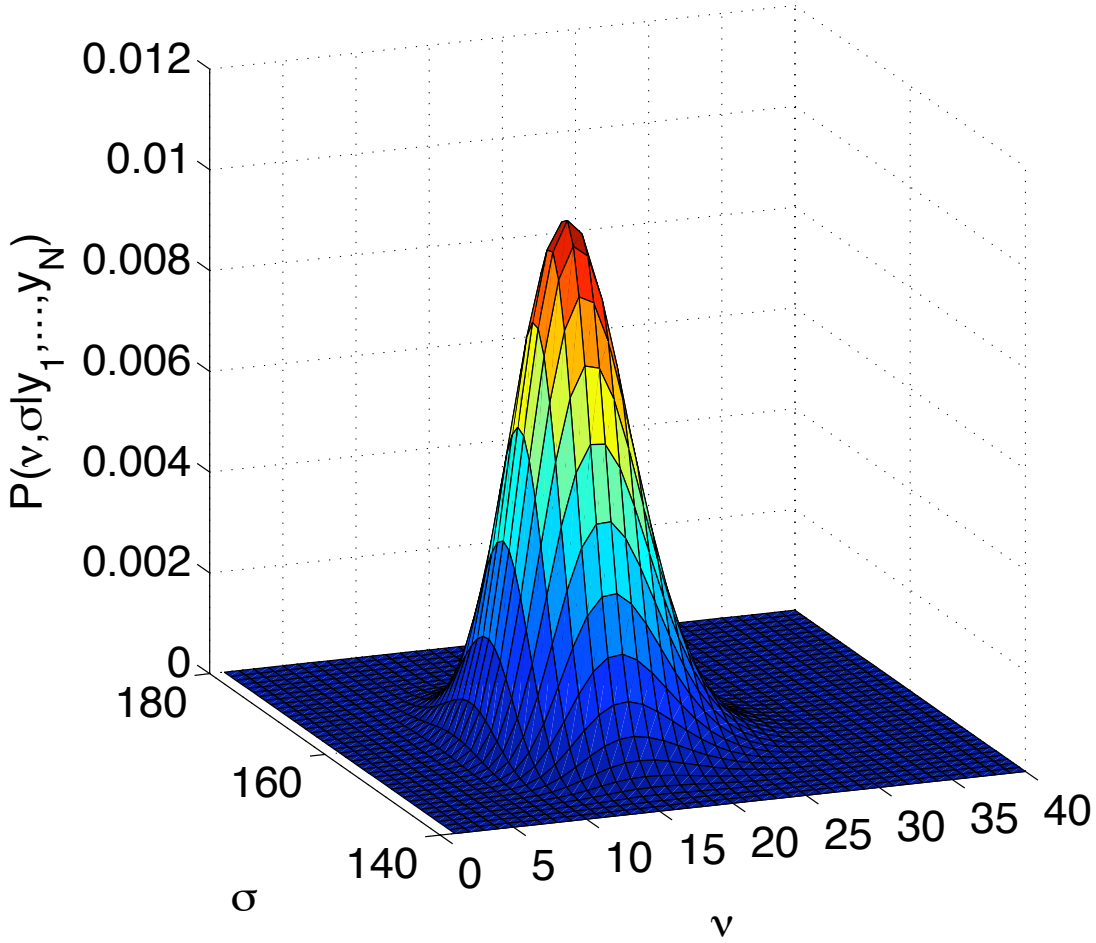
Validation - simulation studies



Validation - simulation studies

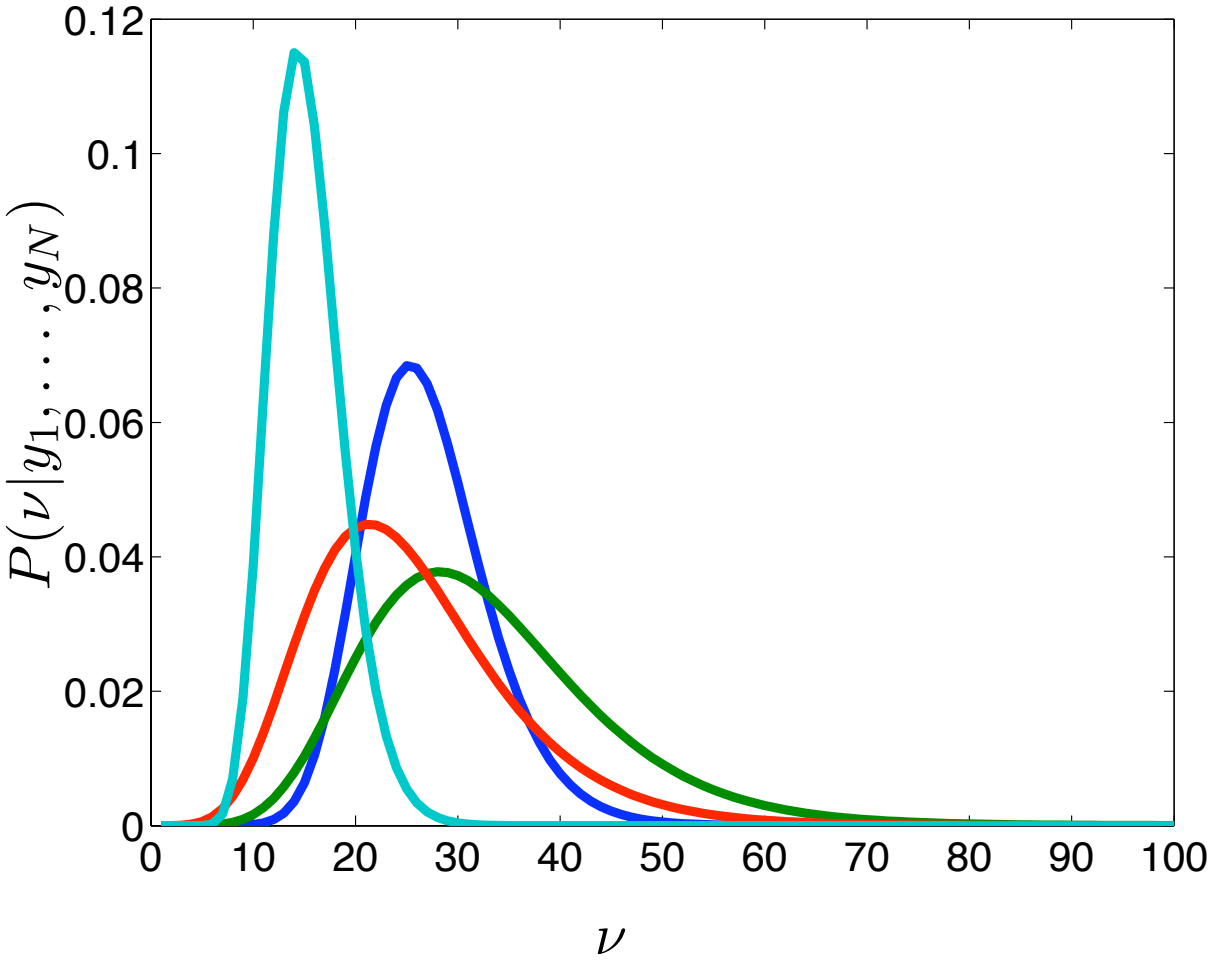


Posterior belief in ν, σ



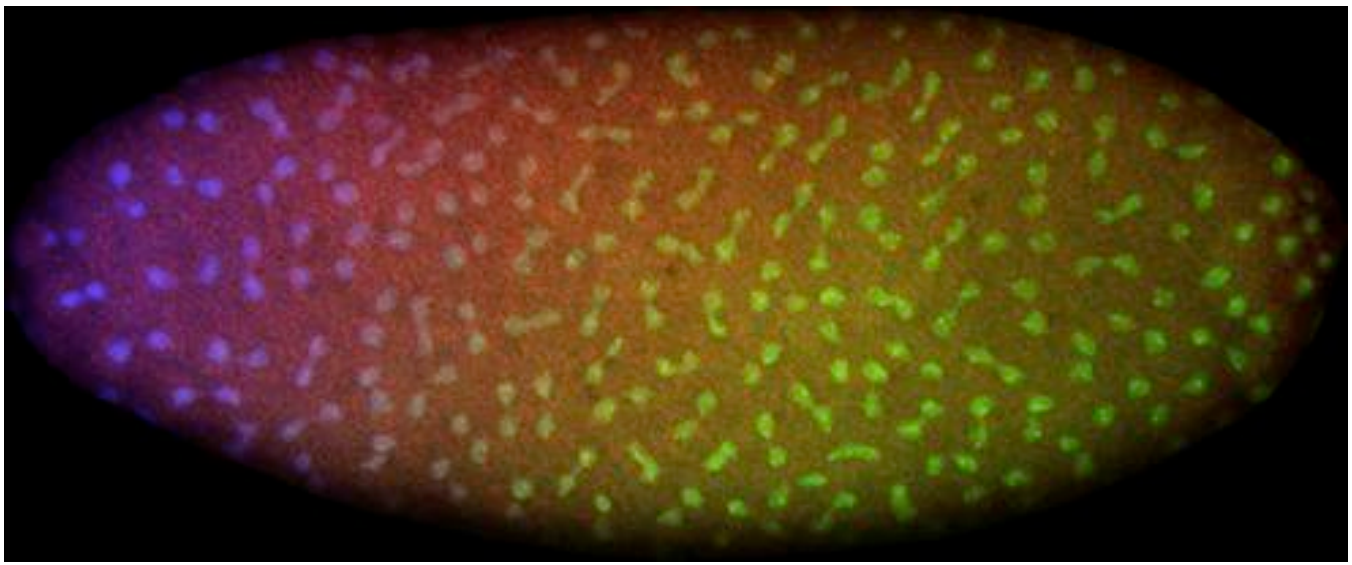
- Most likely ν about 15 (as with simple method!)
- Noise estimated at $\sigma \approx 160$

Consistency across experimental runs



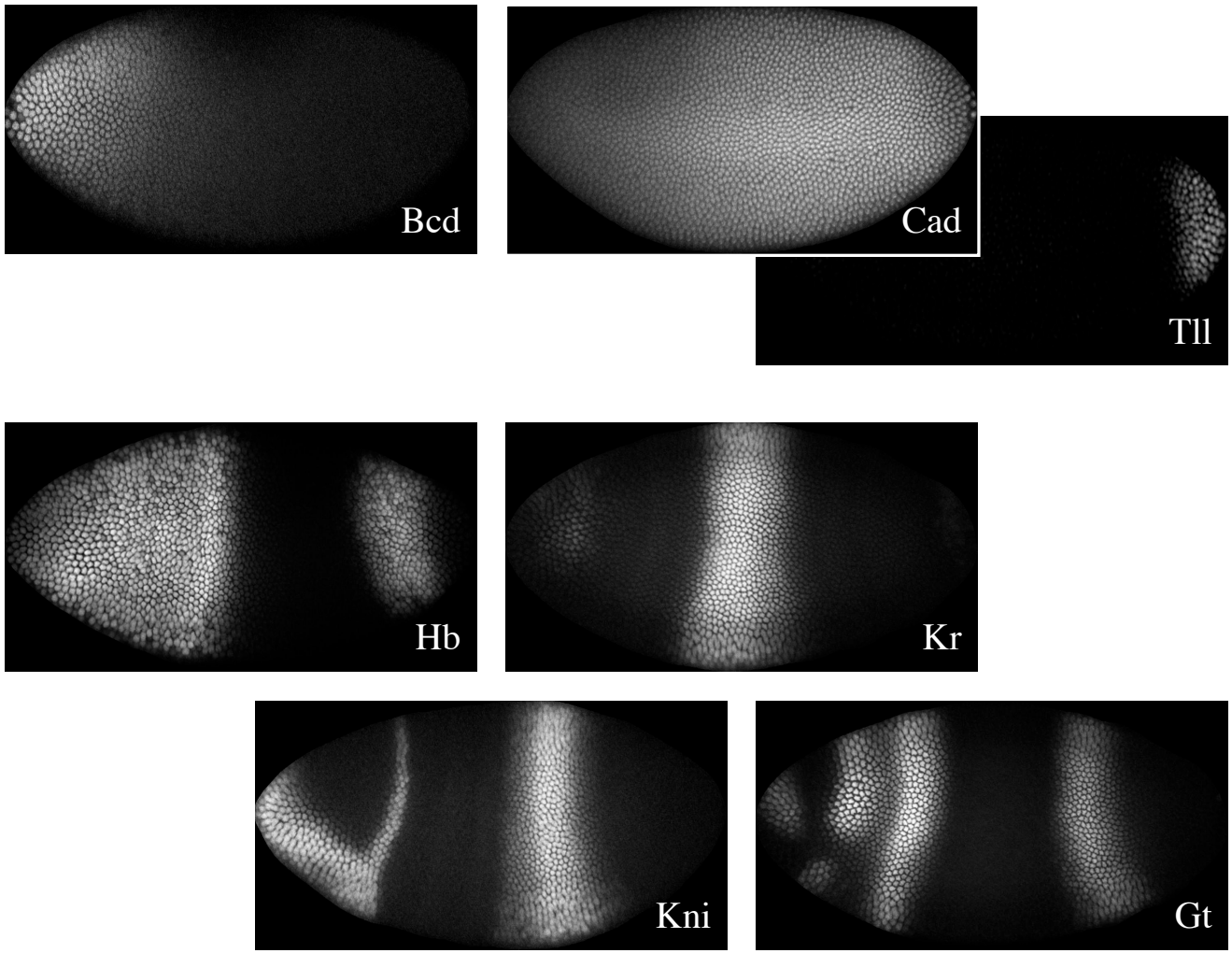
Part I Summary

- Protein copy number can be estimated from variability in fluorescent intensity between daughter cells
- (Though we don't yet have independent confirmation)
- The Bayesian (complicated) approach gives estimates of uncertainty in parameters
- Similar work in progress for *Drosophila*:

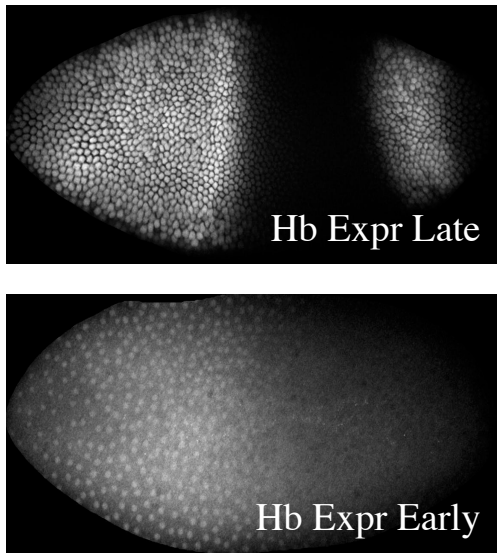


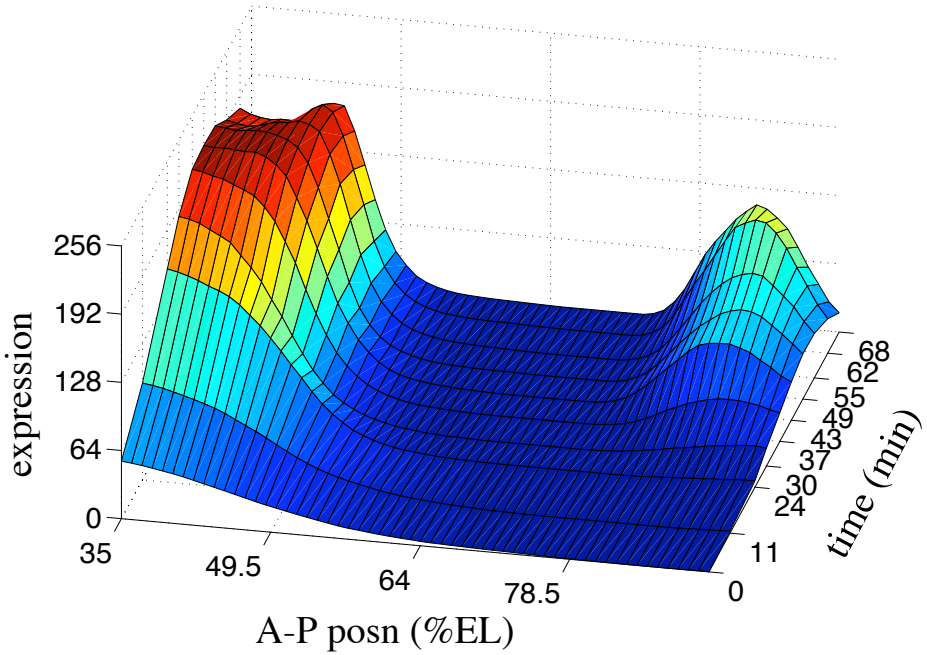
Part II: Modeling regulatory interactions in the gap gene system of *Drosophila melanogaster*

The problem: Modeling the gap gene network in fruit flies



The data


Hb Expr Late
Hb Expr Early


expression
256
192
128
64
0
35
49.5
64
78.5
0
A-P posn (%EL)
11
24
30
37
43
49
55
62
68
time (min)

Previous work by Reinitz and colleagues (Jaeger et al. 2004a,b)

Let $v^a(x,t)$ be the expression of gene a (protein) at time t and anterior-posterior position x.

A PDE model for protein levels:

$$\frac{\partial v^a(x,t)}{\partial t} = \underbrace{P^a(v(x,t))}_{\text{production}} - \underbrace{\gamma^a v^a(x,t)}_{\text{decay}} + \underbrace{D^a \frac{\partial^2 v^a(x,t)}{\partial x^2}}_{\text{diffusion}}$$

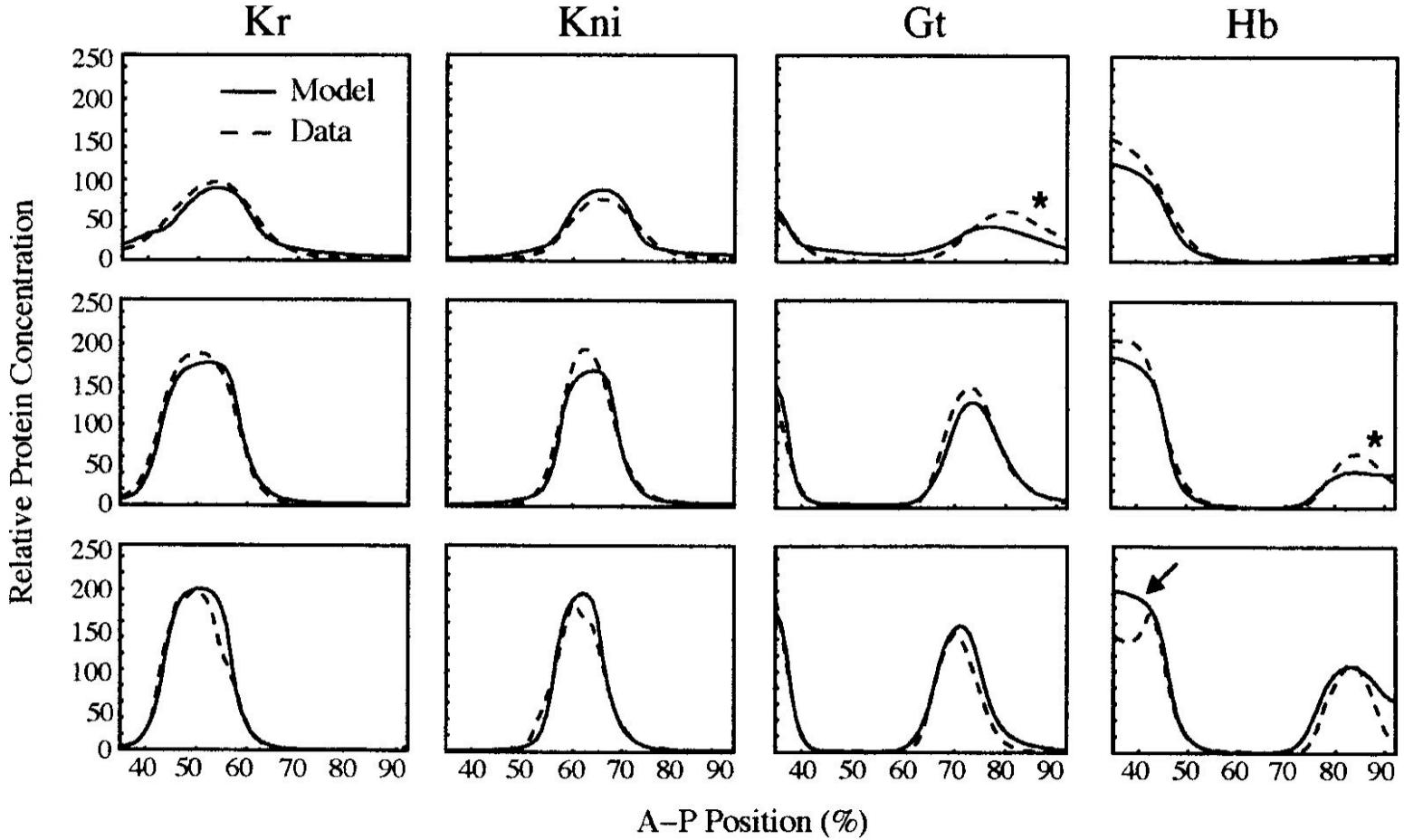
where

$$P^a(v(x,t)) = R^a g\left(\sum_b T^{ab} v^b(x,t) + h^a\right)$$

where $g(u) = \frac{1}{2}\left(\frac{u}{\sqrt{u^2+1}} + 1\right)$ is sigmoidal.

Fitted models

They fit all parameters using multiple runs of a parallel simulated annealing algorithm.



The problem?

- □ 2 CPU-years fitting time! ($\approx$ 2 months on their 10-node parallel processor)
- □ Did not explicitly test RPJ network structure
- □ Did not check sensitivity to various modeling assumptions

$\Rightarrow$ Optimization is hard for the usual reasons:

  - parameter dependencies
  - plateaus and local minima in error surface

Fitting Differential Equations

- Suppose we have time series data $x_o(t)$ for $t = t_1, t_2, \ldots, t_N$.
- Postulate an ODE model of the form $\dot{x}(t) = f(x, \theta)$, where f is a dynamics function parameterized by θ.
- There are two main classes of criteria one might optimize when fitting a differential equation:
  1. Trajectory-based error: $E_{traj} = \sum_t \|x_o(t) - x(t)\|^2$ where $x(t)$ is the solution to the ODE model (from some initial condition)
  2. Derivative-based error: $E_{deriv} = \sum_t \|\hat{\dot{x}}_o(t) - f(x_o(t), \theta)\|^2$ where $\hat{\dot{x}}_o(t)$ estimates the time derivatives of the data at time t.

Differences between E_{traj} and E_{deriv}

E_{traj}:

- Minimizes the difference between simulated and observation expression
- Error is typically highly nonlinear in model parameters
- Even evaluating error requires solution of ODE model

E_{deriv}:

- Minimizes the difference between estimated time derivatives and modeled time derivatives
- Error evaluation is simple
- Error minimization is a regression problem
- Error is typically much less nonlinear in model parameters
- Behaves poorly when data is noisy (though there are ways of fixing that — see functional data analysis)
- But. . . when simulated, model may not match observed expression well

A hybrid solution for the original problem

$$\frac{\partial v^a(x,t)}{\partial t} = R^a g\left(\sum_b T^{ab} v^b(x,t) + h^a\right) - \gamma^a v^a(x,t) + D^a \frac{\partial^2 v^a(x,t)}{\partial x^2}$$

1. Estimate $\frac{\partial v^a(x,t)}{\partial t}$ — in fact, we estimate production, decay and diffusion components separately
2. Optimize $R^a, T^{ab}, h^a, \gamma^a, D^a$ so that:

$$P^a_{est}(x,t) \approx R^a g\left(\sum_b T^{ab} v^b_o(x,t) + h^a\right)$$

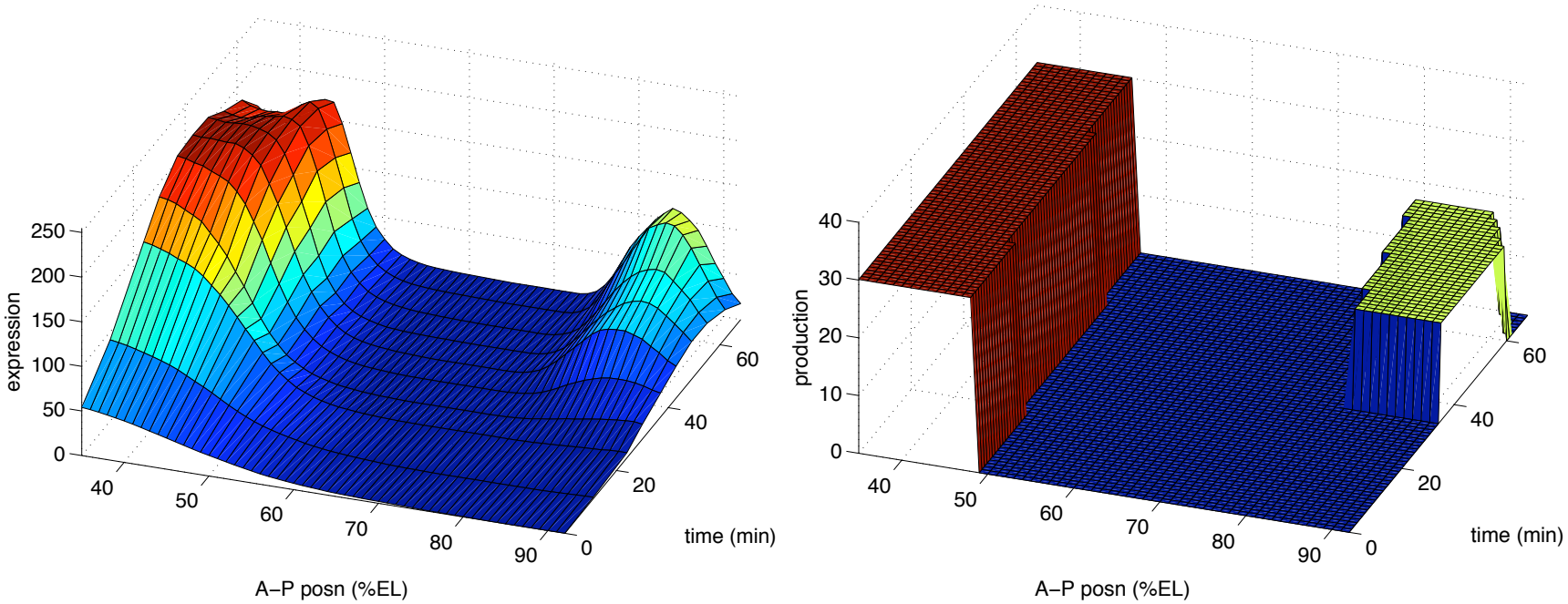
3. Further tune params to minimize trajectory-based error

$\Rightarrow$ Computationally efficient; trajectory-based optimization easy if initial params good enough; in end, model optimized to simulate correctly.

Step 1: Estimate $P^a(x,t), \gamma^a, D^a$

$$\frac{\partial v^a(x,t)}{\partial t} = P^a(x,t) - \gamma^a x^a(x,t) + D^a \frac{\partial^2 v^a(x,t)}{\partial x^2}$$

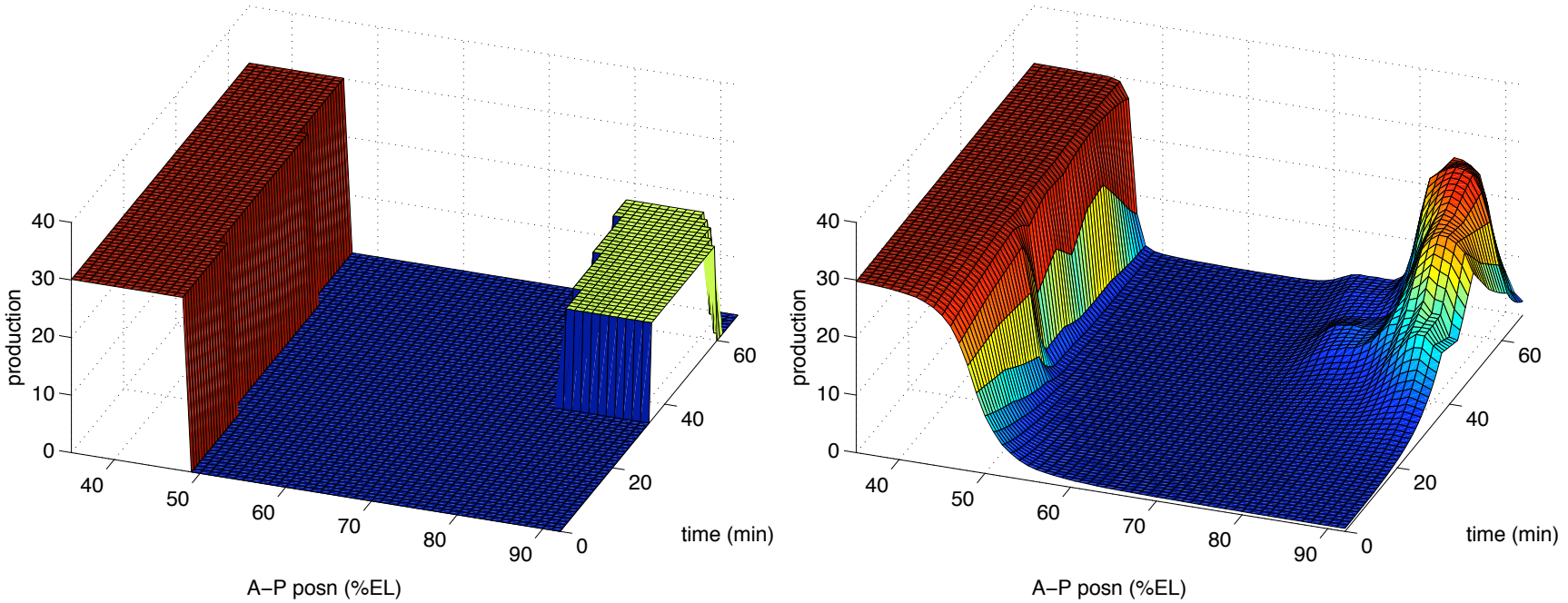
- Production given by quadrilateral patches of space-time
- Optimize so simulated expression matches observed



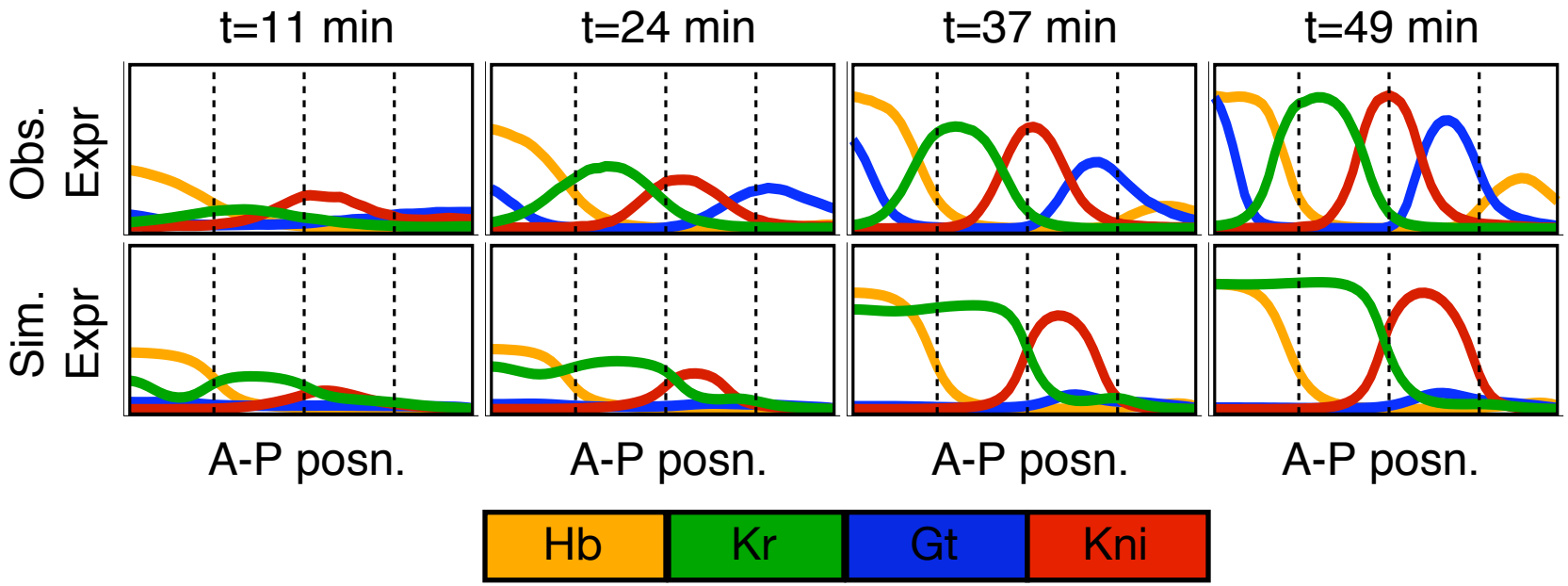
Step 2: Estimate R^a, T^{ab}, h^b based on $P^a(x,t)$

$$P^a_{est}(x,t) = R^a g\left(\sum_b T^{ab} v^b_o(x,t) + h^a\right)$$

- Repeated gradient descent to minimize sum squared error

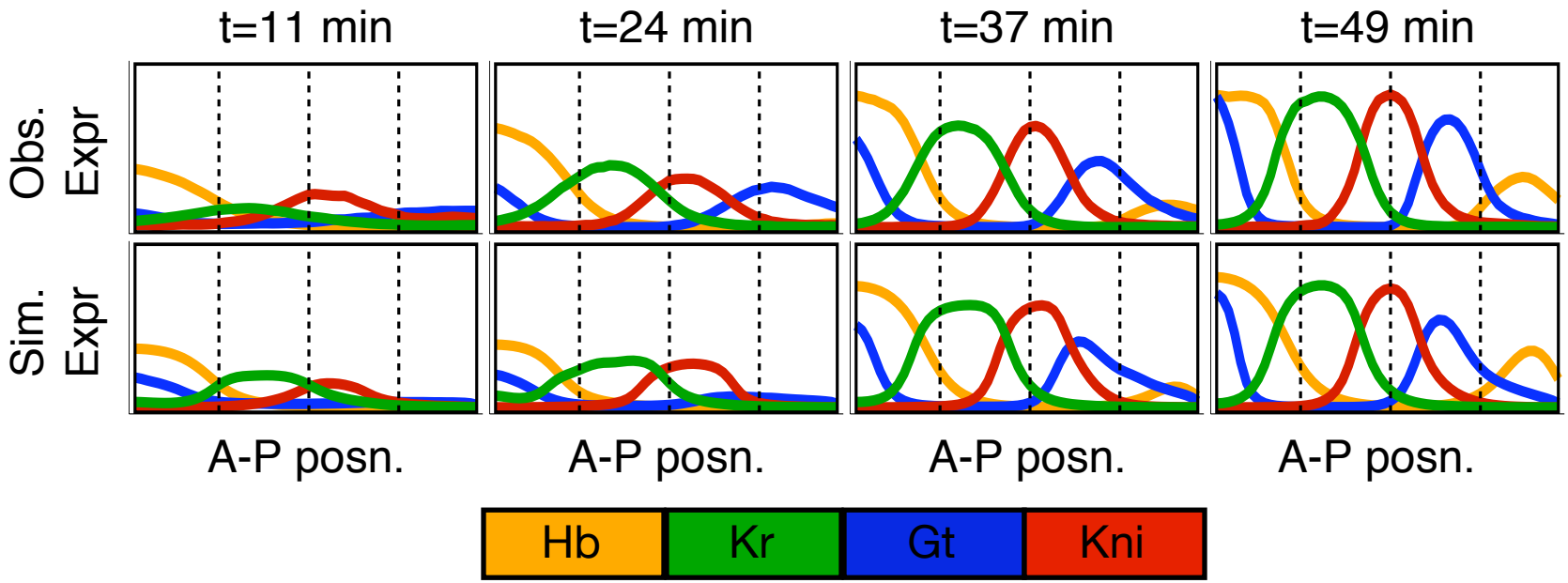


Simulating PDE with $R^a, T^{ab}, h^a, \gamma^a, D^a$ gives poor fit



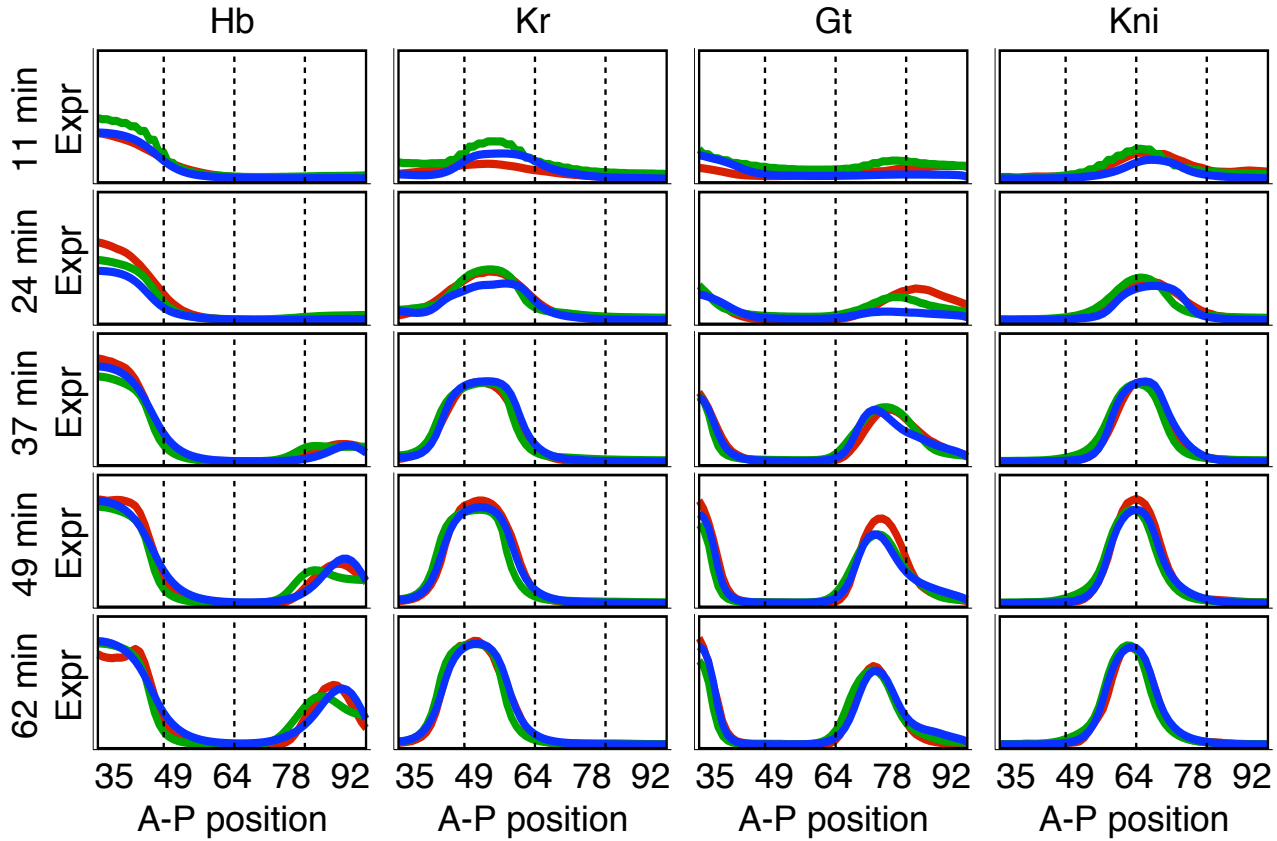
Step 3: Tune $R^a, T^{ab}, h^a, \gamma^a, D^a$ to get good fit

- Repeated stochastic local search



Results

Obtained similar results to Jaeger et al. (2004a,b), in terms of expression and regulatory relationships!

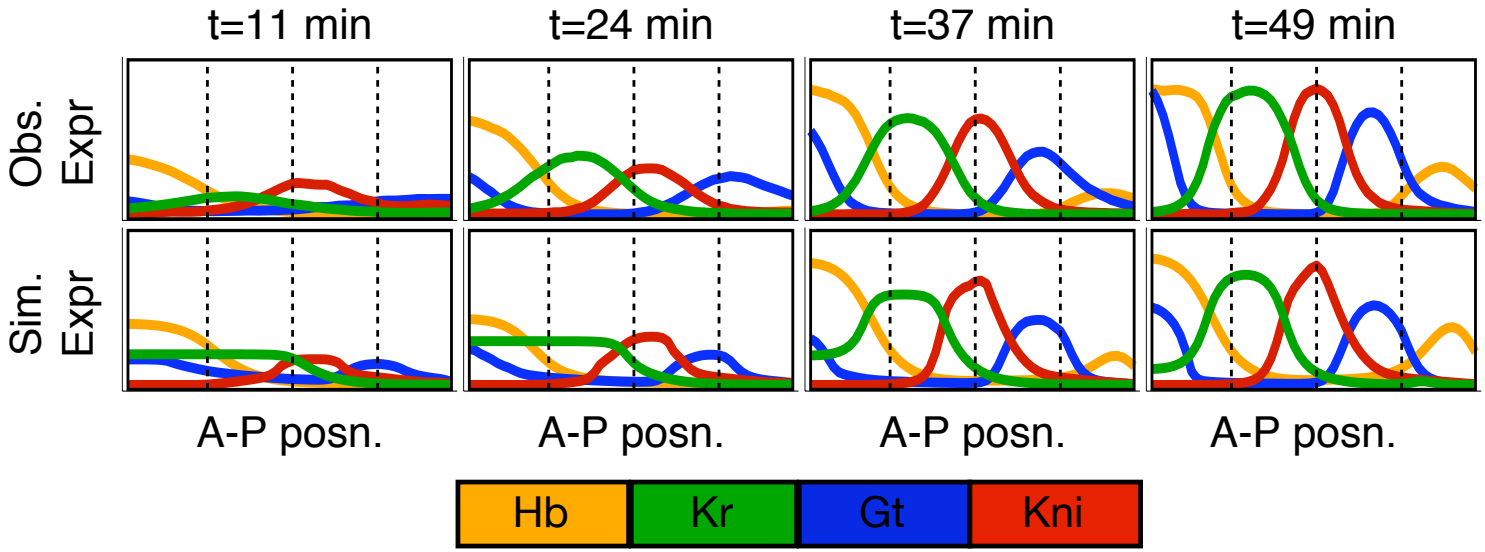


Data (red); Jaeger et al. (green, RMS 12.08); Our fit (blue, RMS 12.29)
36 hours computation

Is that the only regulatory architecture that works?

- Next, we fit a model of the same form but limited to the Rivera-Pomar & Jackle regulatory relationships
- Regulatory weights T^{ab} corresponding to links not in the RPJ model are fixed at zero
- Regulatory weights T^{ab} corresponding to link in the RPJ model are constrained to have the appropriate sign
- A few exceptions:
  - We allowed TII to activate Hb
  - There was an extra negative weight T^{Kr,Hb^2} multiplied by $(v^{Hb}(x,t))^2$, to allow Hb to have a dual regulatory effect on Kr

Model restricted to RPJ structure



RMS error 15.88

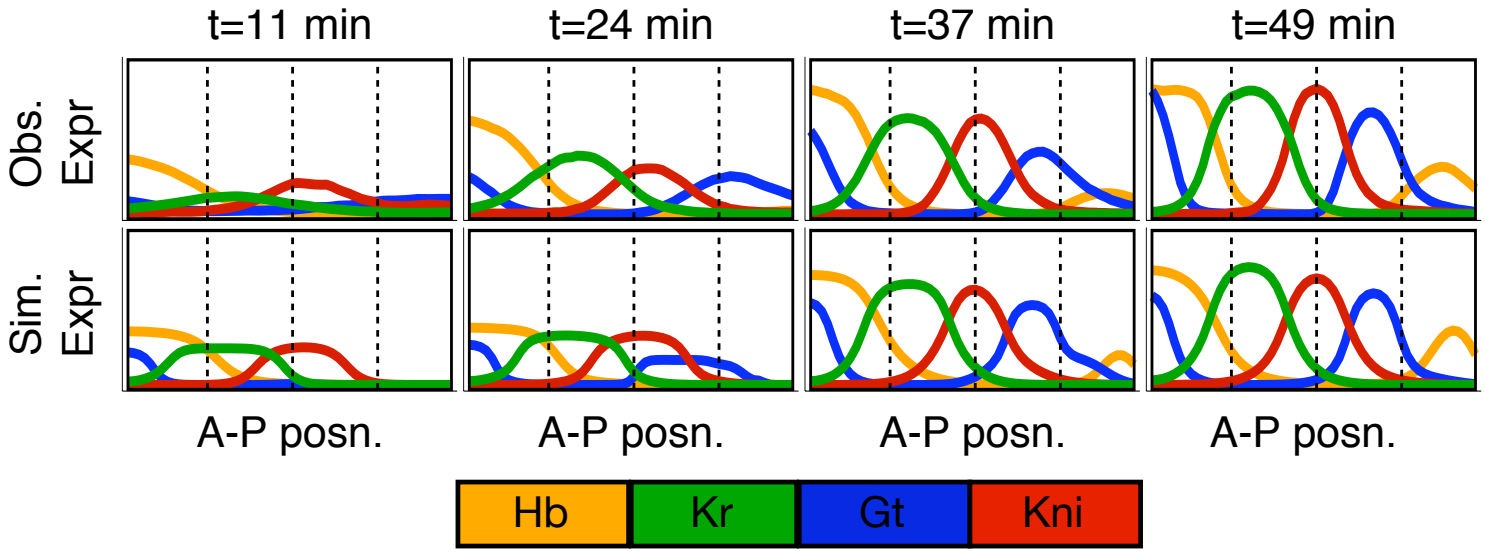
Does the mathematical form of the model matter?

- Next, we fit a piecewise-constant ("logical") model for production
- We assumed production if at least one activator and no repressors exceed thresholds

$$P^{Hb} = \begin{cases} R^{Hb} & \text{if } (v^{Bcd} > 20 \text{ or } v^{Hb} > 90) \text{ and } v^{Kr} < 140 \\ & \text{and } v^{Kni} < 10 \\ 0 & \text{otherwise} \end{cases}$$

- Optimized thresholds, but not structure of network – we borrowed the structure of the first, unconstrained fit

Logical model



RMS error 14.83

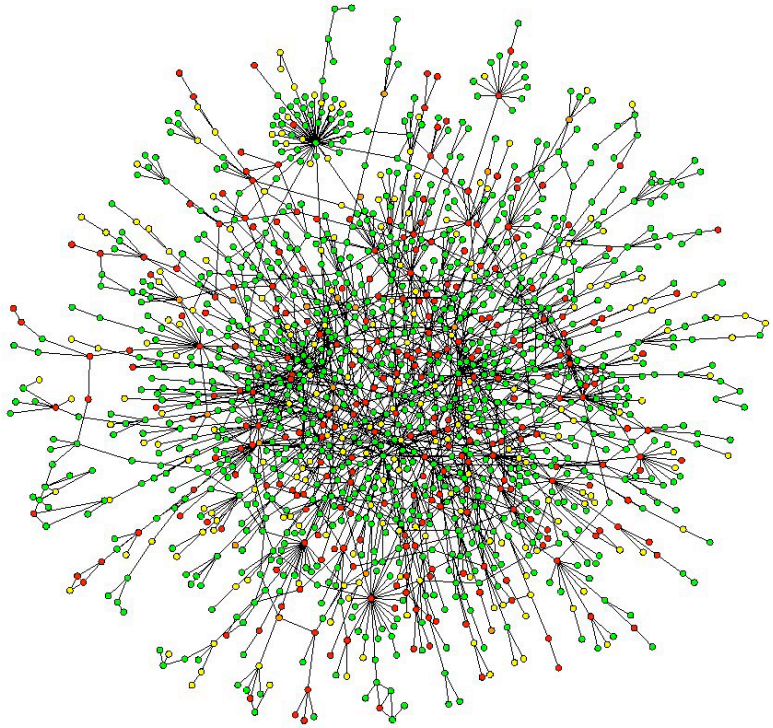
Part II Summary

- A 3-step decomposition of the PDE fitting process
  - Allows estimation of regulatory parameters first by regression
  - Resulting in much faster fitting
  - Relevant to ODE fitting as well
- Future work
  - Theoretical justification for the algorithm
  - Modeling pair-rule genes
  - Quantitative agreement with mutant phenotypes

Part III: Searching for coherent subnetworks in large interaction networks

Large interaction networks

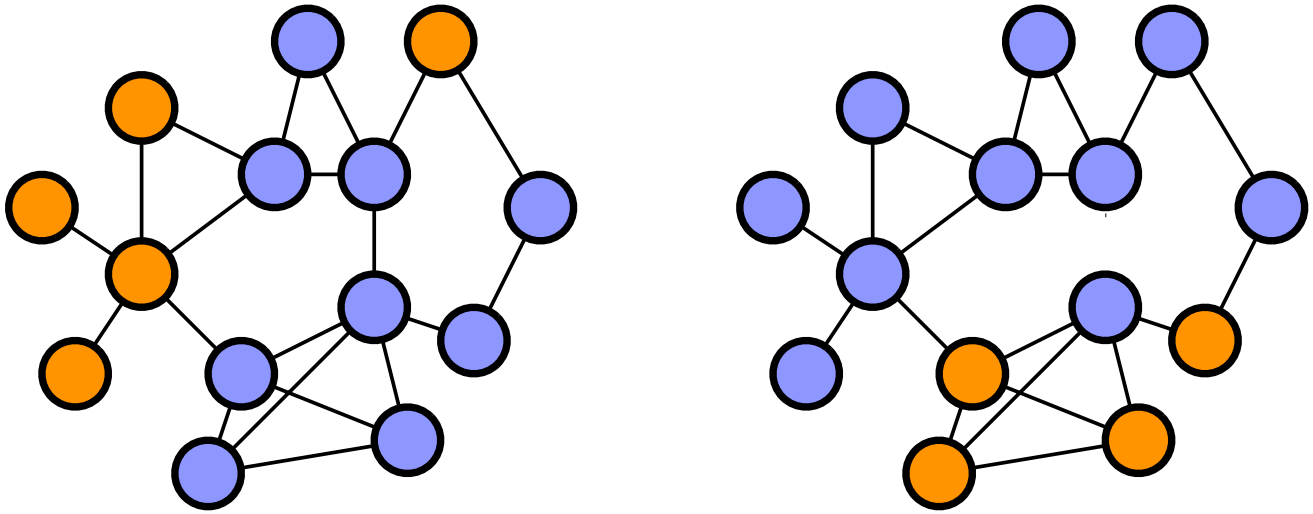
As we learn more about gene and protein interactions on a genome scale, network diagrams become a big, tangled mess! The network does not separate neat subnetworks, because...



- Proteins may have multiple functions or be "promiscuous"
- Subnetworks communicate non-hierarchically
- Links indicate (e.g.) protein-protein or protein-DNA interactions that happens under some conditions

Active subnetworks

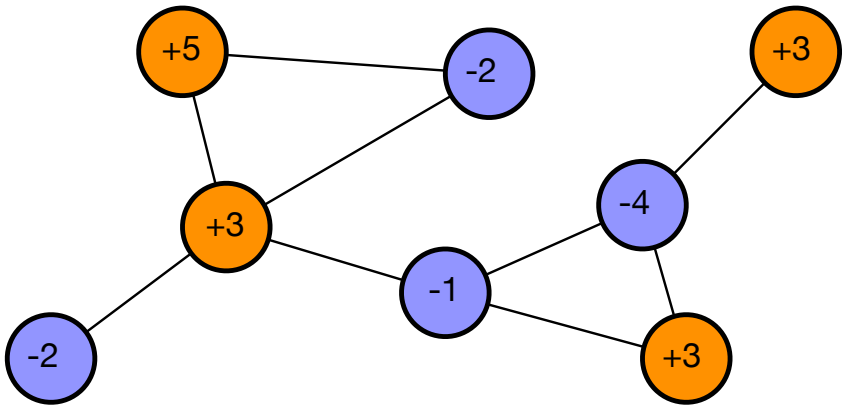
- Different parts of the network may be active at different times / under different conditions



Try to find active subnetworks

(Similar to Ideker *et al.* 2001,2002)

- Weight vertices by evidence for differential expression
- Find connected subsets with high total weight

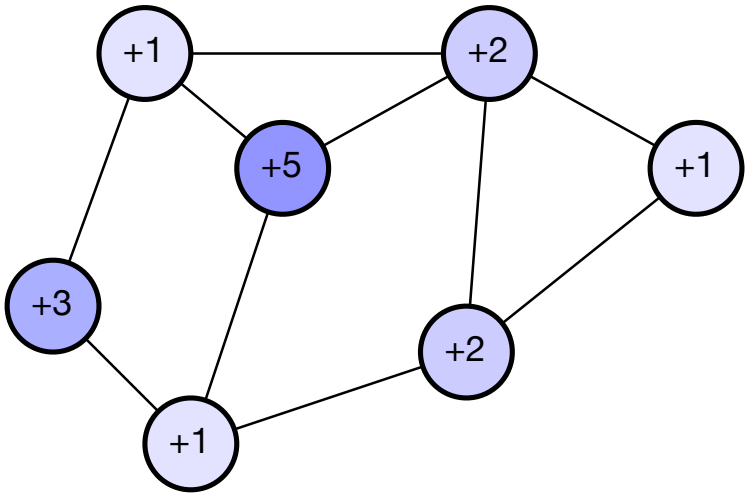


$\Rightarrow$ Unfortunately, this problem is intractable (NP-hard), inapproximable, not f.p.t. You have to check all 2^N subsets.

Steiner trees in a graphs

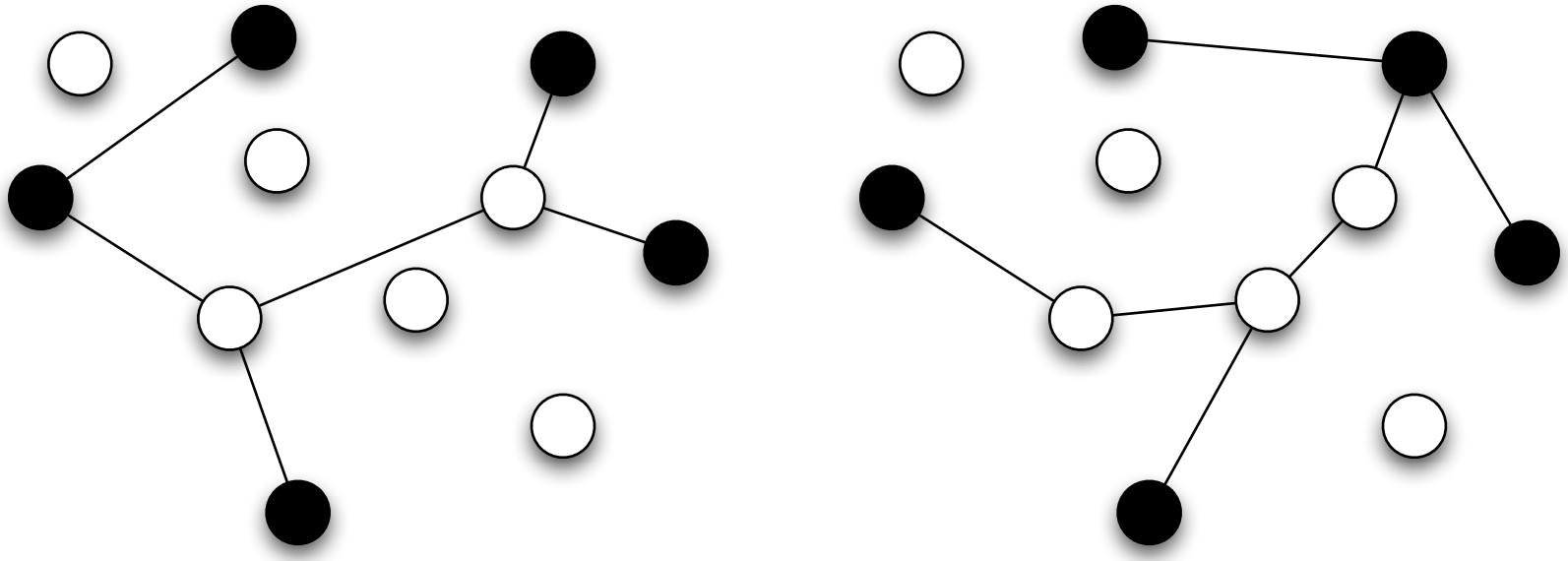
(Scott *et al.* 2005)

- Weight vertices by evidence *against* differential expression
- Being with a seed set of genes S
- Find a connected subset A with $S \subset A$ of minimal total weight



Finding the Steiner tree in a graph

- A modification of the Dreyfus-Wagner algorithm can solve this problem exactly in time $O(3^{|S|}\text{Poly}(N))$, where N is the total number of vertices. (It is a dynamic programming approach, that looks at different ways of breaking Steiner trees into smaller Steiner trees.)



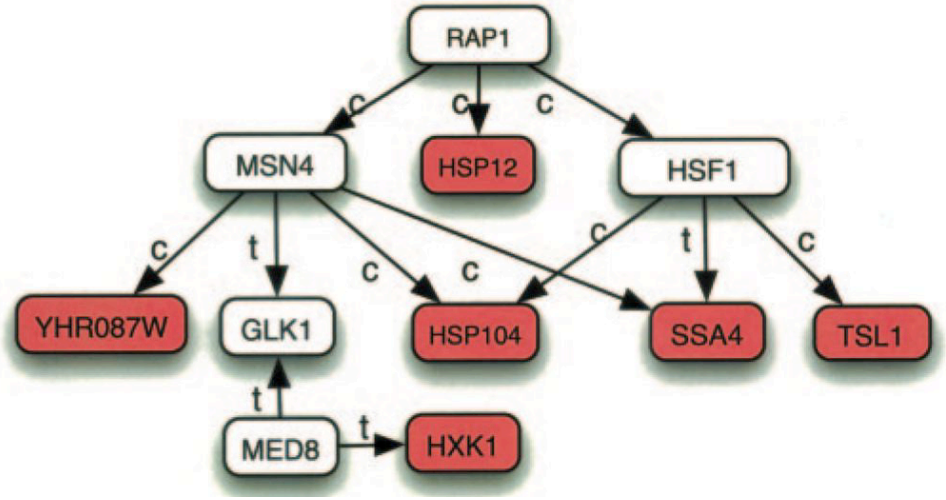
- It is also approximable, and well-solved in practice by heuristics

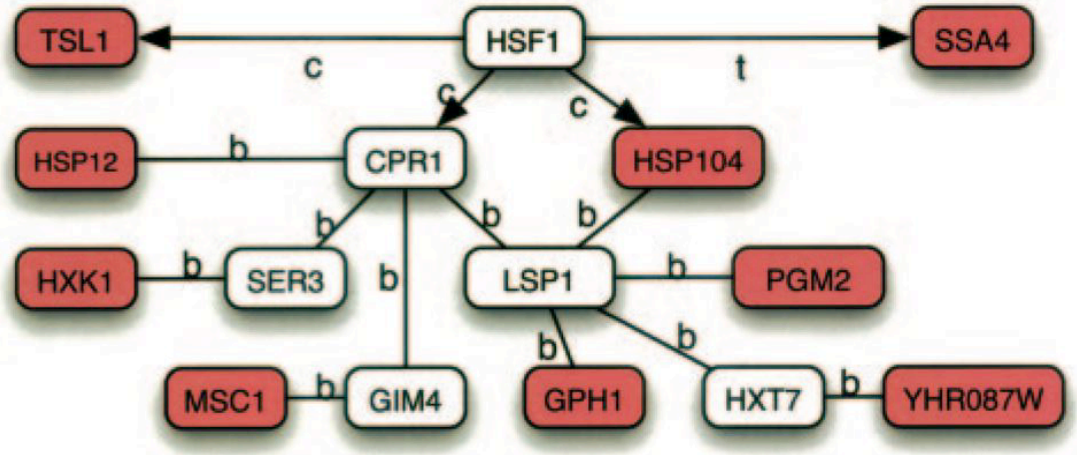
Computational experiments

We defined an interaction network with

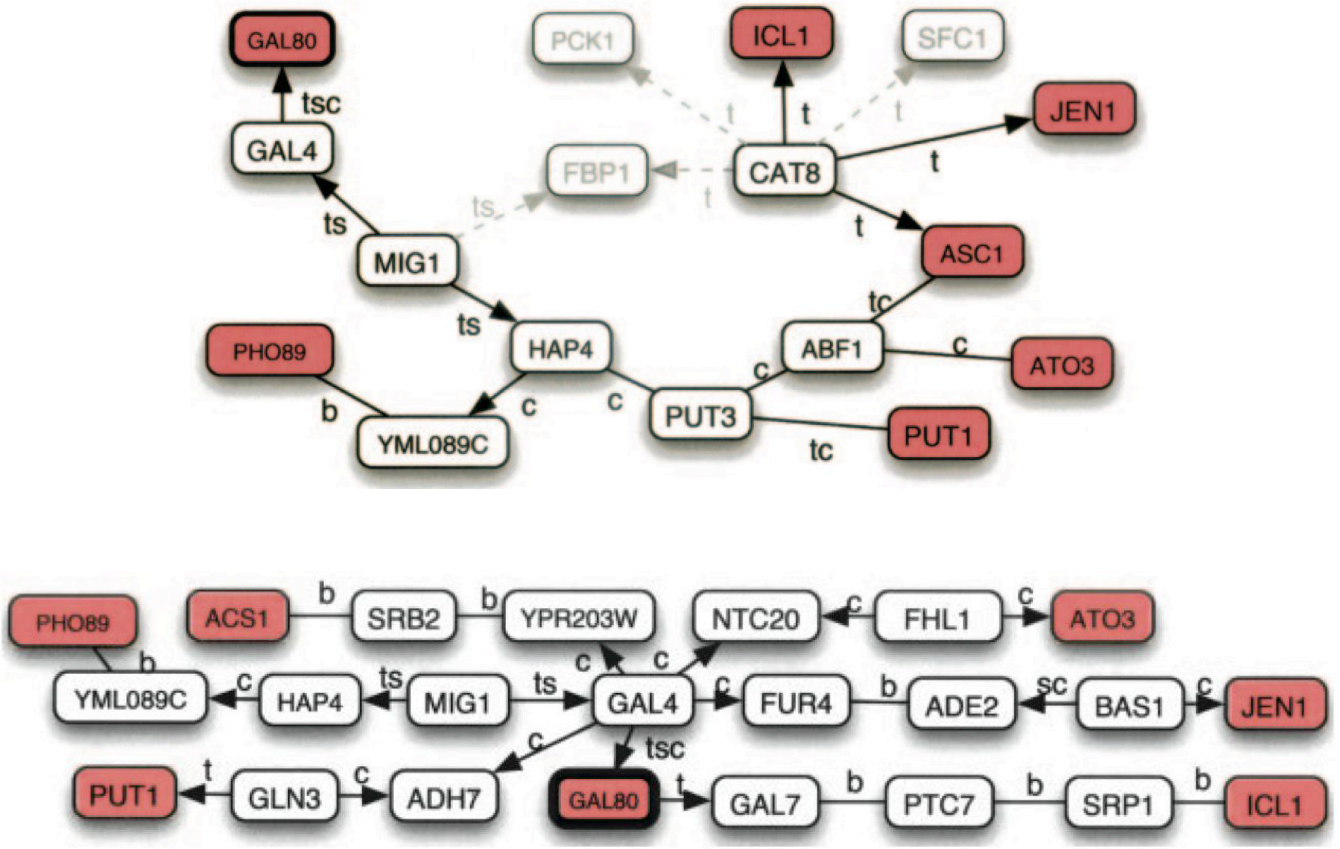
- 5458 vertices corresponding to yeast genes
- 23,642 edges taken from BIND (yeast protein-protein interactions), TRANSFAC, SCPD and a ChiP-Chip data set (yeast protein-DNA interactions)
- Vertex weights indicate differential expression based on microarray experiments

Connecting heat-responsive genes

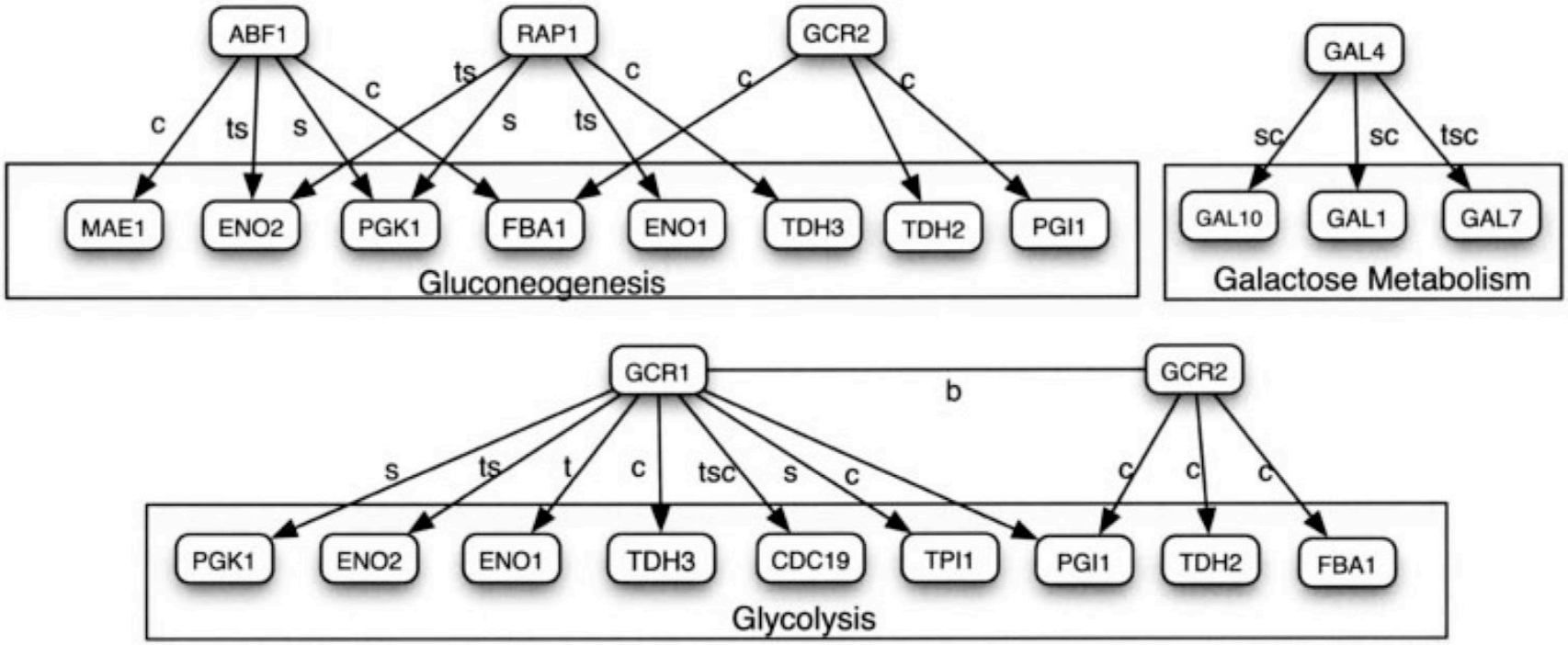




Connecting GAL80 to differentially expressed genes



Connecting co-expressed genes in metabolic pathways



Part III Summary

- We can efficiently find an active subnetwork that connects a seed set of vertices
- Should be considered an exploratory, rather than explanatory, tool
  - Solutions may not be unique
  - Solutions can be sensitive to incorrect vertex weights, missing links
- Links can be weighted too, representing uncertainty about an interaction

The End