The Local Edge Machine (**LEM**): Inference of Dynamic Models of Gene Regulation

Xin Guo
Department of Applied Mathematics,
The Hong Kong Polytechnic University

Joint work with Kevin A. McGoff, Anastasia Deckard, Christina M. Kelliher, Adam R. Leman, Lauren J. Francey, John B. Hogenesch, Steve B. Haase, and John L. Harer

July 17, 2017, FoCM 2017, Barcelona

Inference of dynamic models of gene regulation

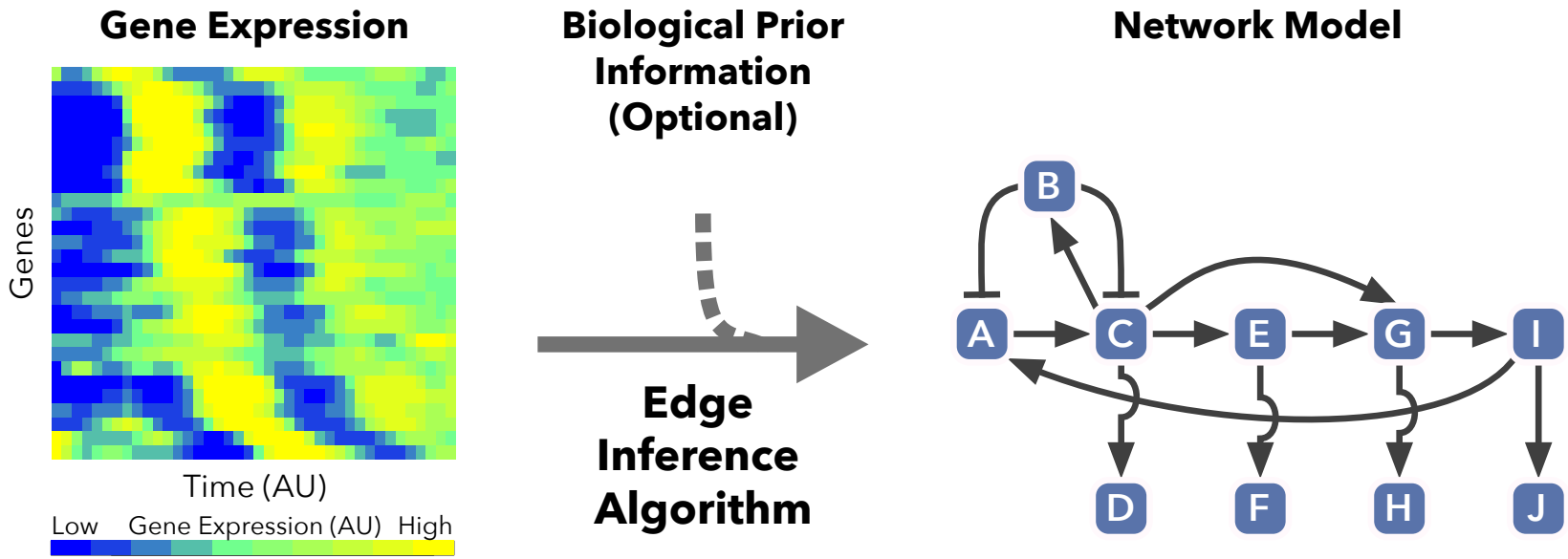


Figure: The inference problem

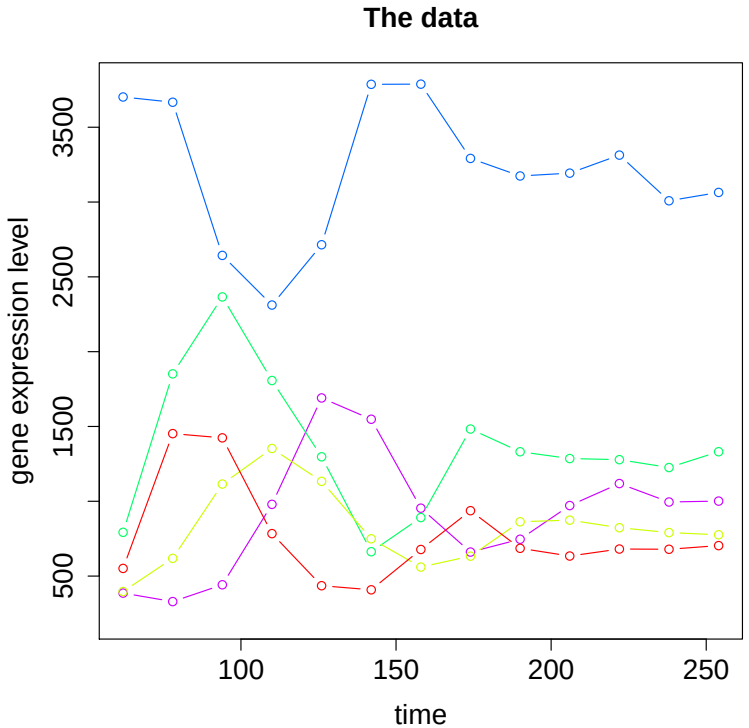
Motivations

1. Using time-course transcriptome data to infer the structure of transcription networks is considered a major problem in computational biology.
2. Practicing systems biologists continue to rely on manual curation of network models.

pictures from http://www.bear.org, http://en.wikipedia.org, and Lennart Nilsson/Scanpix

Challenges

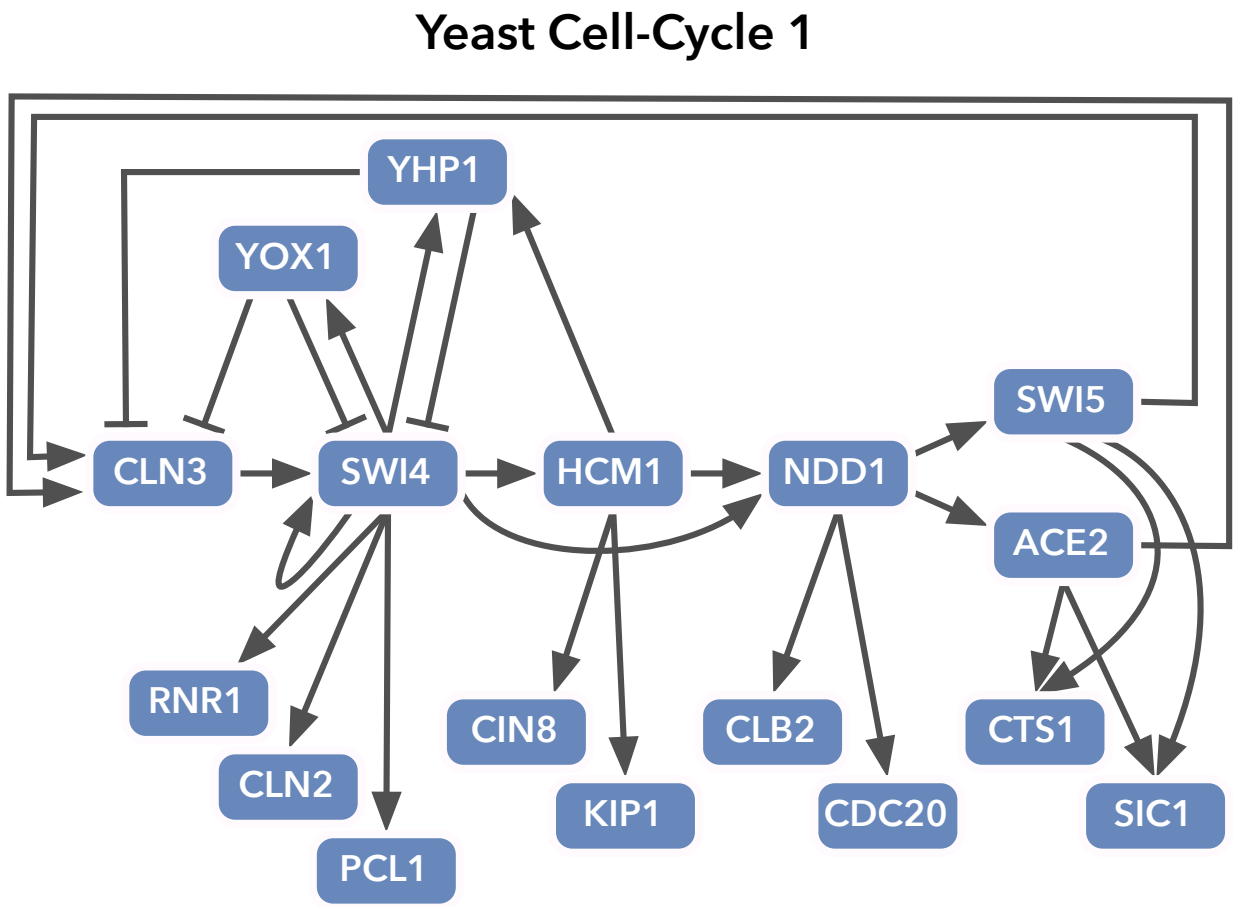
1. low sampling rate, dampened[1], and unknown noise scale



[1] Orlando et al. Cell Cycle 6:4, 478-488, 2007.

Challenges

2. complicated network structure, large hypothesis space



Related works

- boolean/discrete networks, Kauffman, Nature, 1969; Liang & Han, BMC Systems Biology, 2012; Perkins & Hallett & Glass, J. Theor. Biol., 2004; Perkins & Wilds & Glass, Phil. Trans. R. Soc. A, 2010.
- calibration of gene expression data, Sun et al., Ann. Appl. Stat., 2014; Reilly et al., JASA, 2003.
- deconvolution of gene expression data, Orlando et al., Ann. Appl. Stat., 2009; Orlando et al., Cell Cycle, 2007.
- sparsity, Wu et al., JASA, 2014 (LASSO with additive model); "Inferelator", Bonneau et al., Genome Biology, 2006, and Greenfield et al., Bioinformatics, 2013.
- reactiondiffusion PDE, Perkins et al., PLoS Comput. Biol., 2006.
- mutual information, "TD-ARACNE", Zoppoli et al., BMC Bioinformatics, 2010; "Granger Causality", Granger, Econometrica, 1969.
- gene clustering and motif discovery, Gupta & Ibrahim, JASA, 2007.
- analysis of gene regulation involved in the immune response, Heard et al., JASA, 2012.
- parameter estimation for differential equations from statistics, Qi & Zhao, AoS, 2010; Papavasiliou & Ladroue, AoS, 2011; Xun et al., JASA, 2013.

Hill function model

Denote $X = X(t)$, $Y = Y(t)$ the expression levels of two genes X and Y. We model the regulation of gene X towards Y as

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = \gamma - \beta Y + \alpha r(X),$$

for repression and

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = \gamma - \beta Y + \alpha a(X),$$

for activation, respectively. Here

$$r(X) = \frac{K^n}{X^n + K^n}, \text{ and } a(X) = \frac{X^n}{X^n + K^n},$$

are the Hill functions.

- K:half-maximal activation/repression level which suggests the concentration of regulator needed for half-maximal transcription
- α: maximal transcription rate
- n: Hill coefficient, which controls the steepness of the Hill function
- γ: basal synthesis rate
- β: basal degradation rate

Multiple regulations

We model complex regulation of one gene Y by up to three simultaneous regulators, where multiplication of Hill functions is interpreted as an "AND" gate and addition as an "OR" gate.
In general, one expects biological networks to be sparse[2], and even in cases where this assumption is broken, we may seek to identify the most dominant components of a regulation

• When two regulators A and B regulate C in "AND" gate, then their regulating effect is a product

$$\alpha_{A,B,C} h(x_A, K_{A,C}, n_{A,C}) h(x_B, K_{B,C}, n_{B,C}).$$

• When A and B regulate C in "OR" gate, then Hill functions are summed

$$\alpha_{A,C} h(x_A, K_{A,C}, n_{A,C}) + \alpha_{B,C} h(x_B, K_{B,C}, n_{B,C}).$$

• Repressors dominate. When one regulator acts as a repressor, it is combined with all the other regulators in "AND" gate.

[2]Yeung et al., PNAS, 2002; Gardner et al., Science, 2003

Local Edge Machine (LEM): the model

We consider a gene regulatory network $\mathcal{N} = \{X_1, \ldots, X_N\}$.
- $X_i(t)$: expression level of X_i at t
- D: data, $\{X_i(t_j)\}_{i=1,\ldots,N;j=1,\ldots,T}$
- $l_i = (j, h)$: logical regulatory information. $j = 1, \ldots, M$, $h \in \{\text{activate}, \text{repress}\}$
- $\mathcal{L}$: totality of all the possible regulating types. $\{l_i\}_{i=1}^{N} \subset \mathcal{L}^N$, $|\mathcal{L}| = 2N$
- $f(X, l, \theta, \mathbf{X}(t))$: regulating function of $X \in \mathcal{N}$ corresponding to the logic l and parameter θ. Here $\mathbf{X}(t) = (X_1(t), \ldots, X_N(t))$.

For example, if $l = (j, \text{activate})$ then

$$f(X, l, \theta, \mathbf{X}(t)) = \alpha \frac{X_j^n}{X_j^n + K^n} + \gamma - \beta X,$$

where $\theta = (n, K, \alpha, \beta, \gamma)$.

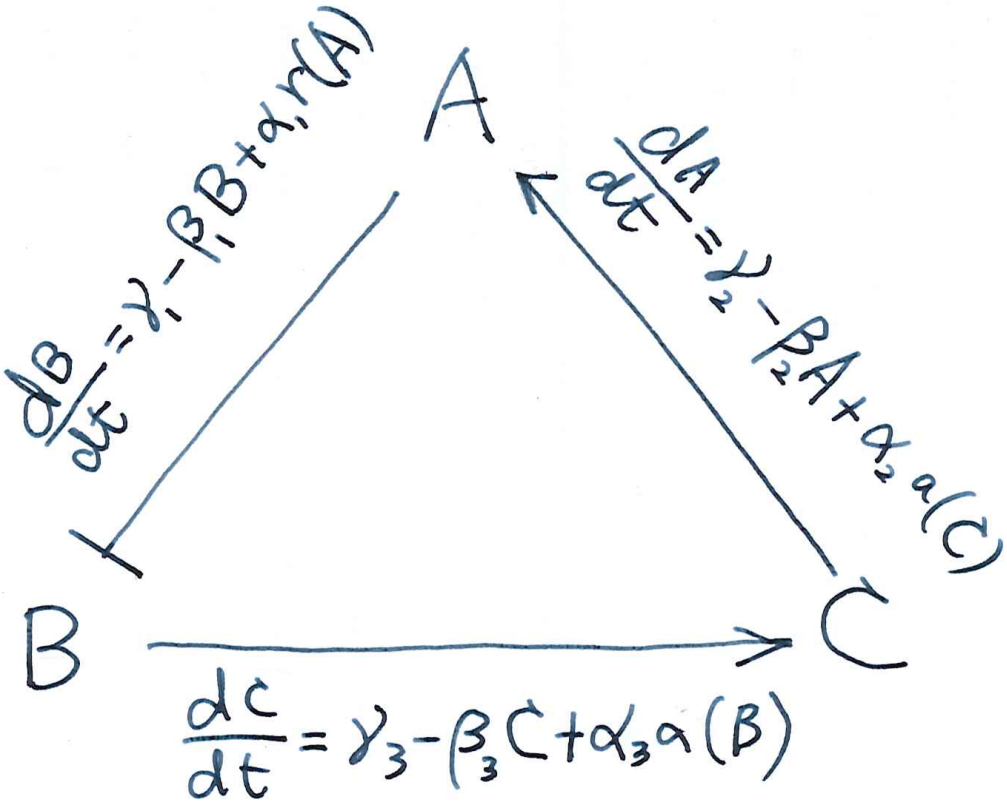
LEM: the model, continued

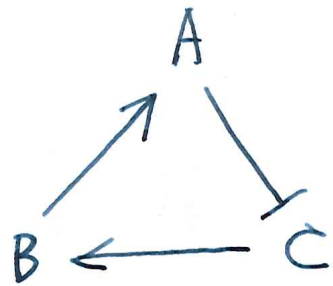
For a specific set $\{l_i\}_{i=1}^N$ of logics, and a set $\{\theta_i\}_{i=1}^N$ of parameters, the LEM model is a differential equation system $\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}, \{l_i\}, \{\theta_i\}, \mathbf{X})$, or specifically,

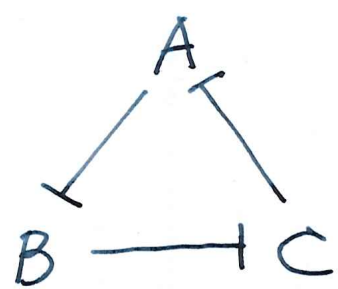
$$\frac{\mathrm{d}X_i(t)}{\mathrm{d}t} = f(X_i, l_i, \theta_i, \mathbf{X}), \quad i = 1, \ldots, N.$$

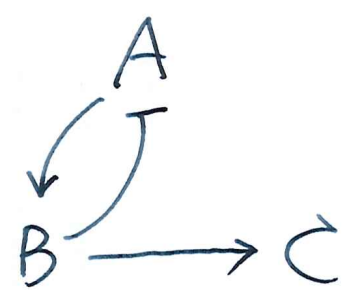
We see that fitting the LEM model is a parameter estimation problem for differential equations. This kind of problem is extensively studied in the literature of statistics[3].

[3] Qi & Zhao, AoS, 2010; Papavasiliou & Ladroue, AoS, 2011; Xun et al., JASA, 2013.


A
B
C
$\frac{dB}{dt} = \gamma_1 - \beta_1 B + \alpha_1 r(A)$
$\frac{dA}{dt} = \gamma_2 - \beta_2 A + \alpha_2 a(C)$
$\frac{dC}{dt} = \gamma_3 - \beta_3 C + \alpha_3 a(B)$


A
B
C


A
B
C


A
B
C

Current strategies for estimating differential equation parameters

1. Discretization methods[4], of which the data fitting process is usually referred to as nonlinear least squares[5], where a least square risk function is minimized. A numerical ODE solver is used to compute the risk function value and its gradient (via sensitivity differential equations). Drawbacks: computationally intensive; inaccurate due to stiffness. LEM somehow belongs to this category yet our novel design overcomes both of the drawbacks.

2. Collocation methods[6], which uses the linear combination of a set of basis functions to approximate the solution.

[4] Biegler et al., AIChE J., 1986.
[5] Wu, AoS, 1981; Malinvaud, Ann. Math. Stat., 1970.
[6] Ramsay et al., JRSSB, 2007.

LEM: nonlinear least squares

For $i = 1, \ldots, N$, define function F_i on $\{t_j\}_{j=1}^T$ by

$$F_i(t_j) = f(X_i(t_j), l_i, \theta_i, \mathbf{X}(t_j)),$$

and extend F_i to the whole interval $[t_1, t_N]$ by linear interpolation. Set

$$\hat{X}_i(t) = \int_{t_1}^{t} F_i(s)\,\mathrm{d}s.$$

Define

$$\ell_i(D, l_i, \theta_i) = \min_{c \in \mathbb{R}} \frac{1}{T} \sum_{j=1}^{T} (X_i(t_j) - \hat{X}_i(t_j) - c)^2.$$

Simple calculus gives a closed form for the gradient of ℓ_i with respect to θ_i without solving any ODE. This makes the computation very efficient. Our simulation shows that simply minimizing ℓ_i over l_i and θ_i suffers heavily from over-fitting.

LEM: (marginal) Gibbs posterior

For $l \in \mathcal{L}$, denote $\Theta(l)$ the set of all the proper parameters θ that fit the logic l. Using the Gibbs posterior principle we obtain that the posterior distribution on the model $(l_i, \theta_i) \in \bigsqcup_{l \in \mathcal{L}} \Theta(l)$, given the data D, is

$$p(l_i, \theta_i | D) = \frac{\exp\left(-\ell_i(D, l_i, \theta_i)\right) \pi(l_i, \theta_i)}{\sum_{l \in \mathcal{L}} \int_{\theta \in \Theta(l)} \exp\left(-\ell_i(D, l, \theta)\right) \pi(l, \mathrm{d}\theta)},$$

therefore we obtain the marginal Gibbs posterior,

$$p(l_i | D) \propto \int_{\Theta(l_i)} \exp\left(-\ell_i(D, l_i, \theta)\right) \pi(l_i, \mathrm{d}\theta).$$

The prior distribution is chosen to be relatively uninformative,

$$\pi(l, \mathrm{d}\theta) = \frac{1}{r} \frac{\mathrm{d}\theta}{\mathrm{Vol}(\Theta(l))},$$

where r is the number of possible logical regulatory relationships, and $\mathrm{Vol}(\Theta(l))$ is the Lebesgue volume of $\Theta(l)$.

Computing Gibbs posterior using Laplace approximation

Let

$$\theta^* \in \arg\min_{\theta \in \Theta(I_i)} \ell_i(D, I_i, \theta).$$

One has

$$\begin{aligned} p(I_i|D) &\propto \int_{\Theta(I_i)} \exp\left(-\ell_i(D, I_i, \theta)\right) \pi(I_i, \mathrm{d}\theta) \\ &= \exp\left(-\ell_i(D, I_i, \theta^*)\right) \int_{\Theta(I_i)} \exp\left\{\ell_i(D, I_i, \theta^*) - \ell_i(D, I_i, \theta)\right\} \pi(I_i, \mathrm{d}\theta) \\ &\approx \frac{\exp\left(-\ell_i(D, I_i, \theta^*)\right)}{2^{d^*}} \sqrt{\frac{(2\pi)^{\dim(\Theta(I_i))}}{\det H}}, \end{aligned}$$

where H is the Hessian of the function $\theta \mapsto \ell_i(D, I_i, \theta)$ at θ^* and d^* is the number of extreme parameters in θ^*.

Prior information enhancing LEM performance

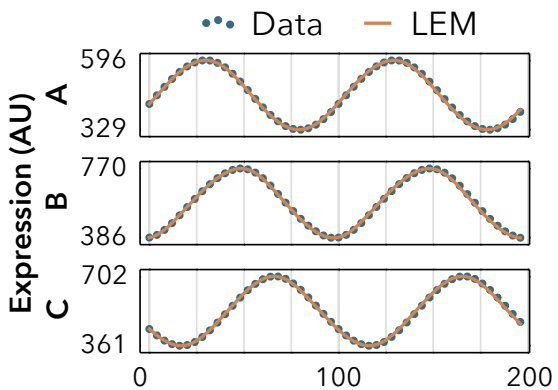
- "gene A never represses others" $\Rightarrow$ set $p(l_A \ni \text{repress}) = 0$
- "gene B is a light bulb" $\Rightarrow$ set $p(l_B \ni \text{repress}) = p(l_B \ni \text{activate}) = 0$.

Here light bulb means a gene that does not regulate other genes.

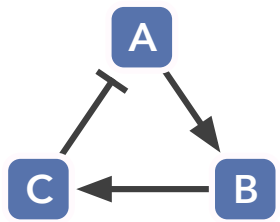
LEM on small networks

i. Gene Expression | **ii. LEM Network & Parameters** | **iii. Network**

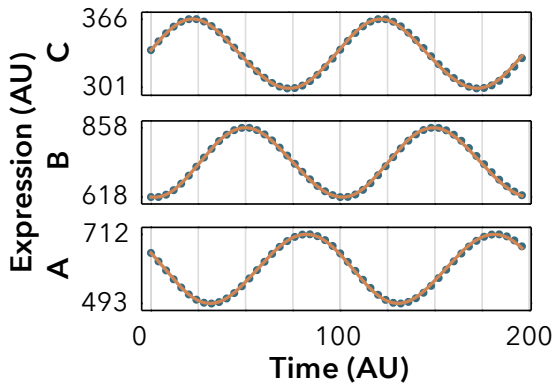
I. *In Silico* 1



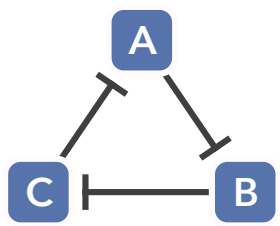
	Reg	γ	β	α	K	n
Real	C ⊣ A	0.398	0.058	43.57	594.72	3.74
	A → B	0.892	0.029	34.22	468.54	6.38
	B → C	0.004	0.026	36.21	651.53	4.76
LEM	C ⊣ A	0.000	0.059	44.28	597.10	3.71
	A → B	1.134	0.029	33.75	467.41	6.55
	B → C	0.143	0.026	35.68	647.96	4.86



II. *In Silico* 2



	Reg	γ	β	α	K	n
Real	C ⊣ A	0.069	0.096	98.58	354.34	5.48
	A ⊣ B	0.660	0.026	36.78	600.22	5.11
	B ⊣ C	0.201	0.012	8.53	711.39	6.40
LEM	C ⊣ A	0.000	0.096	98.62	354.93	5.51
	A ⊣ B	0.000	0.026	37.89	603.82	4.98
	B ⊣ C	0.213	0.012	8.50	713.19	6.44



Comparing with other statistical methods

Network	# Nodes	LEM (AUC)	Inferelator[7] (AUC)	LEM (MCC)	TD-ARACNE[8] (MCC)
In silico 1	3	1	0.9	1	0
In silico 6	3	1	0.8111	1	0
In silico 2	3	1	0.5666	1	0
In silico 7	3	1	0.8555	1	0
In silico 3	5	0.99	0.7857	0.7378	0.4528
In silico 8	5	0.9867	0.6395	0.8261	0.2955
In silico 9	5	1	0.6688	1	0.4
In silico 10	5	1	0.9444	1	0.3931
In silico 4	10	0.9183	0.7105	0.5881	0.0647
In silico 11	10	0.9099	0.6976	0.6342	0.1732
In silico 12	10	0.9064	0.6391	0.7294	0.1381
In silico 13	10	0.8836	0.6614	0.5691	0.0636
In silico 14	10	0.9181	0.6654	0.6934	0.2088
In silico 15	10	0.9237	0.7275	0.6934	0.2292
In silico 16	10	1	0.6552	1	0.3041
In silico 17	10	1	0.8578	1	0.186
In silico 18	10	0.8884	0.5541	0.7462	0.0635
In silico 19	10	0.8824	0.6628	0.3866	0.0391
In silico 5	20	0.878	0.6789	0.5907	0.2146
In silico 20	20	0.9036	0.7718	0.6981	0.2207
In silico 21	20	0.8233	0.6387	0.5285	0.1225
In silico 22	20	0.8157	0.6926	0.4656	0.0524
Yeast cell-cycle 1 (replicate 1)	17	0.8692	0.6705	0.0478	0.0292
Yeast cell-cycle 1 (replicate 2)	17	0.8465	0.6592	-0.0385	-0.0045
Yeast cell-cycle 2 (replicate 1)	8	0.8459	0.6551	0.1732	0
Yeast cell-cycle 2 (replicate 2)	8	0.8404	0.6679	0.0975	0.1091
Yeast cell-cycle 3 (replicate 1)	10	0.7092	0.6064	0.2207	0.0388
Yeast cell-cycle 3 (replicate 2)	10	0.6956	0.6364	0.1459	0.1174
Yeast cell-cycle 4 (replicate 1)	28	0.5138	0.5055	0.0017	0.0307
Yeast cell-cycle 4 (replicate 2)	28	0.4803	0.541	-0.0099	0.0234
Yeast cell-cycle 5 (replicate 1)	19	0.7408	0.579	0.0421	0.0751
Yeast cell-cycle 5 (replicate 2)	19	0.7208	0.6268	-0.0355	0.0744

[7] Bonneau et al., Genome Biology, 2006; Greenfield et al., Bioinformatics, 2013.

[8] Zoppoli et al., BMC Bioinformatics, 2010.

LEM: performance enhanced by prior information

Network	# Nodes	LEM (AUC)	Inferelator (AUC)	LEM (MCC)	TD-ARACNE (MCC)
Yeast cell-cycle 1 (replicate 1)	17	0.9889	0.6705	0.7378	0.0292
Yeast cell-cycle 1 (replicate 2)	17	0.985	0.6592	0.5437	-0.0045
Yeast cell-cycle 2 (replicate 1)	8	0.9681	0.6551	0.6831	0
Yeast cell-cycle 2 (replicate 2)	8	0.9626	0.6679	0.5855	0.1091
Yeast cell-cycle 3 (replicate 1)	10	0.8813	0.6064	0.4452	0.0388
Yeast cell-cycle 3 (replicate 2)	10	0.8778	0.6364	0.4452	0.1174
Yeast cell-cycle 4 (replicate 1)	28	0.8235	0.5055	0.1845	0.0307
Yeast cell-cycle 4 (replicate 2)	28	0.8166	0.541	0.1437	0.0234
Yeast cell-cycle 5 (replicate 1)	19	0.9544	0.579	0.5761	0.0751
Yeast cell-cycle 5 (replicate 2)	19	0.9466	0.6268	0.4405	0.0744

LEM: performance on *in silico* data enhanced by prior information

in silico 16	has 13 Priors						
NoiseLevel	NoPrior	1/6Prior	1/3Prior	1/2Prior	2/3Prior	5/6Prior	FullPrior
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
8	0.9921	0.9938	0.9962	0.9970	0.9980	0.9993	1.0000
16	0.9553	0.9625	0.9683	0.9744	0.9801	0.9843	0.9879
32	0.9158	0.9247	0.9358	0.9463	0.9555	0.9655	0.9737
in silico 18	has 14 Priors						
NoiseLevel	NoPrior	1/6Prior	1/3Prior	1/2Prior	2/3Prior	5/6Prior	FullPrior
0	0.8884	0.8990	0.9102	0.9200	0.9305	0.9400	0.9505
1	0.8881	0.8999	0.9110	0.9219	0.9316	0.9407	0.9516
2	0.8861	0.8970	0.9076	0.9201	0.9313	0.9405	0.9512
4	0.8913	0.9016	0.9099	0.9203	0.9302	0.9397	0.9505
8	0.8366	0.8542	0.8700	0.8873	0.9047	0.9228	0.9372
16	0.8527	0.8684	0.8865	0.8989	0.9159	0.9306	0.9434
32	0.8258	0.8407	0.8586	0.8732	0.8862	0.9026	0.9172
in silico 19	has 14 Priors						
NoiseLevel	NoPrior	1/6Prior	1/3Prior	1/2Prior	2/3Prior	5/6Prior	FullPrior
0	0.8825	0.8944	0.9038	0.9168	0.9241	0.9358	0.9439
1	0.8802	0.8923	0.9050	0.9144	0.9236	0.9344	0.9441
2	0.8911	0.8999	0.9094	0.9184	0.9274	0.9350	0.9426
4	0.8902	0.9002	0.9109	0.9192	0.9291	0.9376	0.9445
8	0.8815	0.8912	0.9002	0.9091	0.9184	0.9280	0.9366
16	0.8475	0.8575	0.8662	0.8725	0.8817	0.8922	0.9005
32	0.8406	0.8508	0.8596	0.8683	0.8795	0.8889	0.8975

LEM: performance on *in vivo* data enhanced by prior information

DATA SET	#Priors	NoPrior	1/6Prior	1/3Prior	1/2Prior	2/3Prior	5/6Prior	FullPrior
Yeast cell-cycle 1 (replicate 1)	25	0.8693	0.8888	0.9108	0.9331	0.9519	0.9733	0.9889
Yeast cell-cycle 1 (replicate 2)	25	0.8465	0.8695	0.8927	0.9163	0.9402	0.9629	0.9854
Yeast cell-cycle 2 (replicate 1)	7	0.846	0.8693	0.8902	0.91	0.9339	0.9504	0.9682
Yeast cell-cycle 2 (replicate 2)	7	0.8404	0.8646	0.8841	0.9055	0.926	0.9441	0.9626
Yeast cell-cycle 3 (replicate 1)	9	0.7092	0.7388	0.7662	0.7985	0.8264	0.8534	0.8832
Yeast cell-cycle 3 (replicate 2)	9	0.6957	0.7246	0.758	0.7866	0.8181	0.8487	0.8776
Yeast cell-cycle 4 (replicate 1)	32	0.5138	0.5664	0.6179	0.6693	0.7196	0.7724	0.8236
Yeast cell-cycle 4 (replicate 2)	32	0.4804	0.5369	0.5915	0.6486	0.7036	0.7596	0.8166
Yeast cell-cycle 5 (replicate 1)	27	0.7409	0.7771	0.812	0.848	0.8843	0.9204	0.9544
Yeast cell-cycle 5 (replicate 2)	27	0.7208	0.7589	0.7959	0.8334	0.8724	0.9101	0.9464

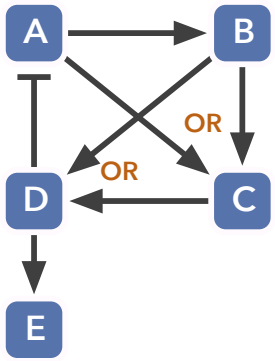
Running time

Network	# Nodes	Time (seconds)	Network	# Nodes	Time (seconds)
In silico 1	3	402	In silico 6	3	456
In silico 2	3	506	In silico 7	3	334
In silico 9	5	1125	In silico 10	5	1028
In silico 3	5	1208	In silico 8	5	1062
In silico 4	10	4454	In silico 11	10	4283
In silico 12	10	3825	In silico 13	10	4467
In silico 14	10	3980	In silico 15	10	4047
In silico 16	10	3999	In silico 17	10	4393
In silico 18	10	3777	In silico 19	10	4379
In silico 5	20	17588	In silico 20	20	16948
In silico 21	20	16843	In silico 22	20	16849
Yeast cell-cycle 1 (replicate 1)	17	9226	Yeast cell-cycle 1 (replicate 2)	17	8628
Yeast cell-cycle 2 (replicate 1)	8	2024	Yeast cell-cycle 2 (replicate 2)	8	1937
Yeast cell-cycle 3 (replicate 1)	10	2949	Yeast cell-cycle 3 (replicate 2)	10	3208
Yeast cell-cycle 4 (replicate 1)	28	24532	Yeast cell-cycle 4 (replicate 2)	28	24515
Yeast cell-cycle 5 (replicate 1)	19	10784	Yeast cell-cycle 5 (replicate 2)	19	11418

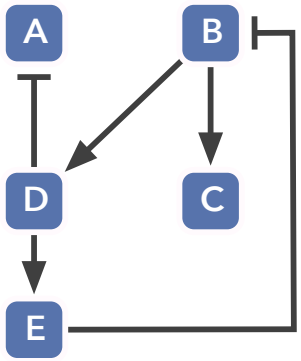
Note that LEM is highly parallelizable. In fact, for solving a network with N nodes, the computation could be decomposed into $2C_RN^2$ independent units (e.g., $C_R = 50$).

Identifiability issue

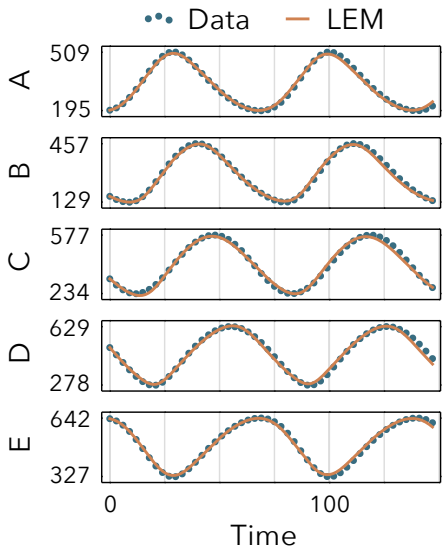
In Silico 3



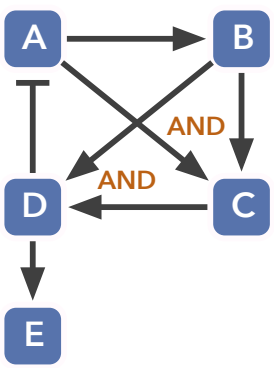
LEM



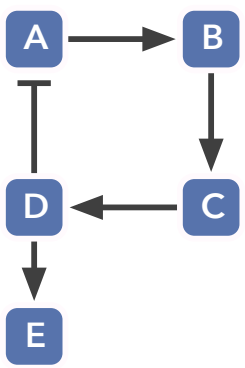
Dynamics



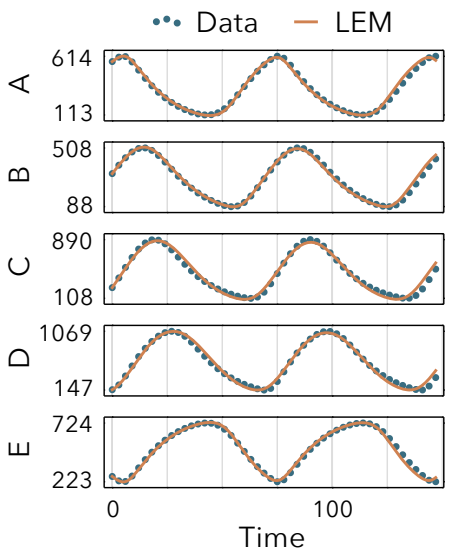
In Silico 8



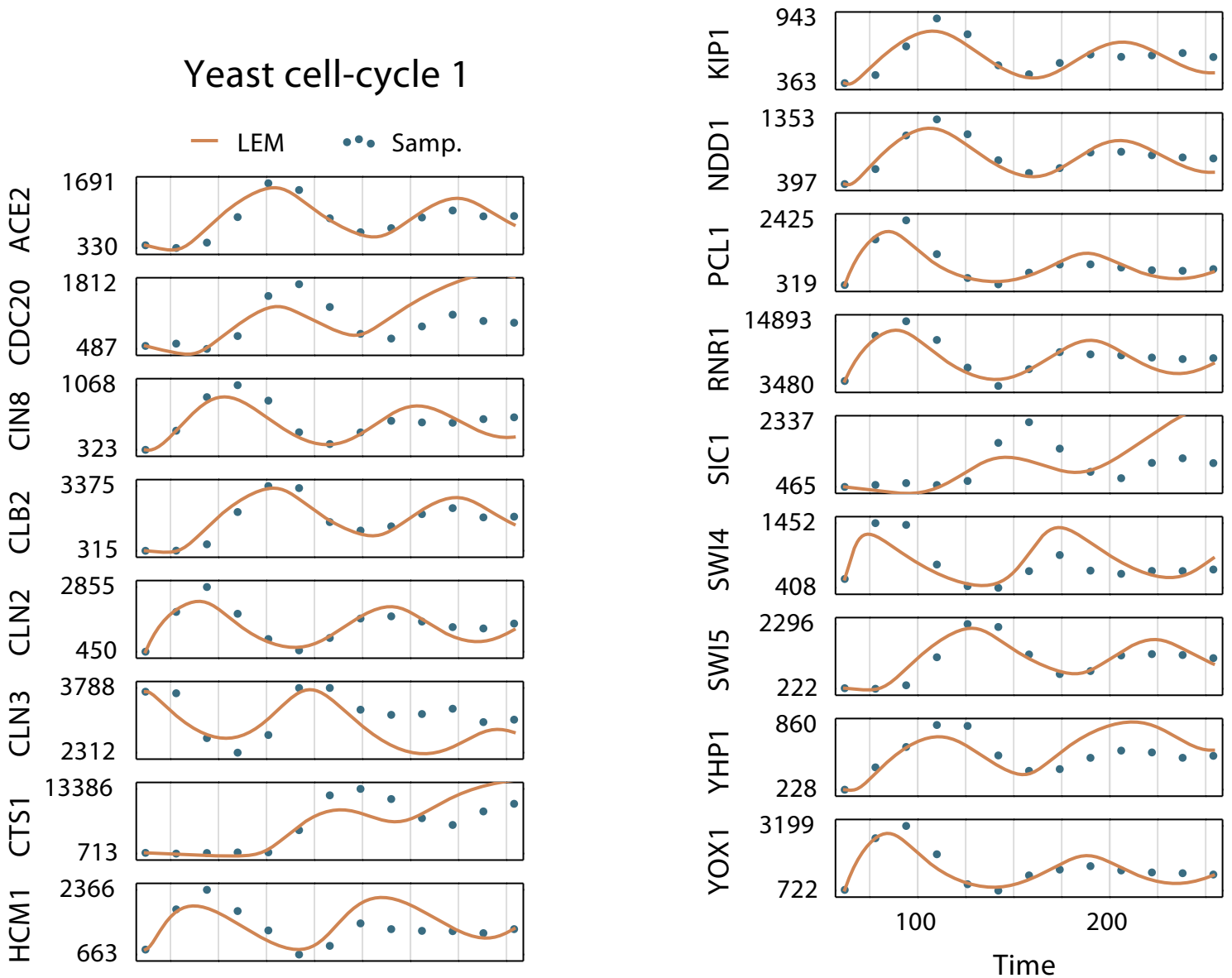
LEM



Dynamics



LEM: inferred transcription kinetics



Conclusions

• LEM is a scalable and precise statistical method to infer regulatory relations from gene expression data. LEM outperforms previously reported statistical methods by wide margins.
• Mathematical and statistical treatments make LEM free from solving differential equations, hence fast.
• The idea of localization makes LEM scalable.
• Large hypothesis space (parameter space of dimension $5N$), together with a proper regularization scheme (the determinant of Hessian on denominator prefers robust networks, similar as the nature), makes LEM precise.
• LEM takes prior information and benefits from it.

- Supported in part by DARPA and NSF
- Supported in part by Research Grants Council of Hong Kong

- Xin Guo, `x.guo@polyu.edu.hk`

Thank you!