Reconstructing Biological Networks using Additive ODE Models

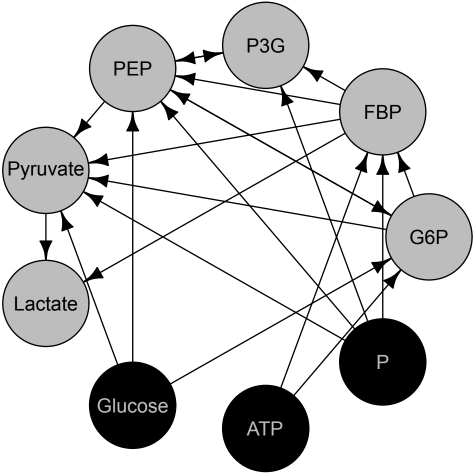
James Henderson
Joint work with George Michailidis
Department of Statistics
University of Michigan

MSSISS March 21, 2014

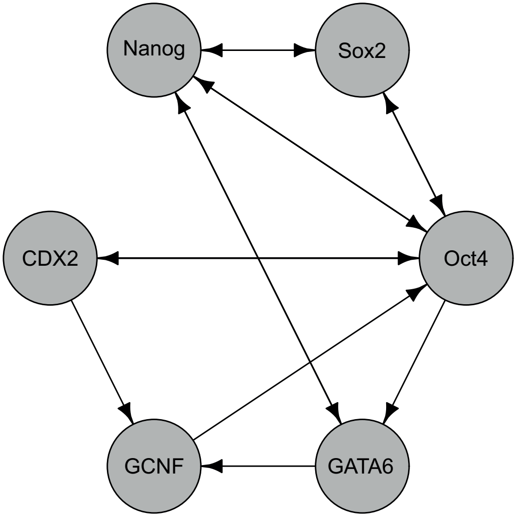
Network Representations of Biological Systems

- Biological processes occur through complex reaction networks involving genes, proteins, metabolites and other biochemical molecules
- Networks provide a compact representation of these processes at an appropriate level of abstraction
- Nodes represent biochemical entities
- Edges connect related entities
- Physical meaning of an edge depends on context

Metabolism: Glycolytic Pathway in Lactocaccus Lactis



Gene Regulation: Mouse Embryonic Stem Cells



Problem and Importance

- Goal: Reconstruct networks using high-throughput data on their nodal entities to determine the edges
- Reconstructing biological networks is a focal problem in systems biology
- Elucidating and understanding the role of networks has many potential applications in basic and applied biology:
  - Metabolic networks help explain how organisms synthesize molecules
  - Gene regulatory networks shed light on how organisms adapt to environmental changes
  - Applications to disease onset, progression, and treatment

Problem

- Goal: Reconstruct networks using high-throughput data on their nodal entities to determine the edges
- We focus on time-series data rather than direct perturbation experiments
  - Time-series data are more readily available
  - There is no clear analogue to a ‘knockout’ in metabolic networks
- Existing approaches include: Vector-Autoregressive Models, Dynamic Bayesian Networks, Process Models specified by ODEs
- Our approach assumes the underlying process can be well approximated by an ODE

Existing Approaches

- Existing approaches include: Vector-Autoregressive Models, Dynamic Bayesian Networks, Process Models specified by ODEs
- Vector-Autoregressive models - assume a linear structure on the level of the trajectories
- Dynamic Bayesian Networks - computationally intractable for even modestly sized networks
- Process Models specified by ODEs

Existing Approaches Based on ODEs

- Most network reconstruction approaches based on ODEs can be viewed as variable selection for the linear model (Oates, 2012).
- Nonlinear approaches usually specify a parametric form for f and then pair parameter estimation with a graph search algorithm (Brunel, 2009).
- Biological processes are often highly nonlinear – even on the level of the derivatives.
- Linear ODEs are a useful but inadequate first approximation.
- Our approach combines nonparametric smoothing with recent advances in ODE estimation to expand the model class.

Formal Problem Statement

- Process model is a dynamic system described by the autonomous first-order differential equation,

$$\dot{x}_1(t) = f_1(x(t)), \quad x_1(0) = x_{01}$$
$$\vdots$$
$$\dot{x}_d(t) = f_d(x(t)), \quad x_d(0) = x_{0d}$$

- More compactly using vectors,

$$\dot{x}(t) = f(x(t)), x(0) = x_0;$$
$$\dot{x}, x : [0,1] \to \mathbb{R}^d;$$
$$f : \mathbb{R}^d \to \mathbb{R}^d.$$

- Our goal is to learn which variables are important in each component of $f(x) = (f_1(x), ..., f_d(x))'$.

Computational Model of Mouse EBSC

$$\dot{x}_1 = \frac{a_0 + a_1 A + a_2 x_1 x_2 + a_3 x_1 x_2 x_3}{1 + b_0 A + b_1 x_1 + b_2 x_1 x_2 + b_3 x_1 x_2 x_3 + b_4 x_4 x_1 + b_5 x_5} - \beta_1 x_1$$

$$\dot{x}_2 = \frac{c_0 + c_1 x_1 x_2 + c_2 x_1 x_2 x_3}{1 + d_0 x_1 + d_1 x_1 x_2 + d_3 x_1 x_2 x_3} - \beta_2 x_2$$

$$\dot{x}_3 = \frac{e_0 + e_1 x_1 x_2 + e_2 x_1 x_2 x_3}{1 + f_0 x_1 + f_1 x_1 x_2 + f_2 x_1 x_2 x_3} - \beta_2 x_3$$

$$\dot{x}_4 = \frac{g_0 + g_1 x_4}{1 + h_0 x_4 + h_1 x_4 x_1} - \beta_4 x_4$$

$$\dot{x}_5 = \frac{i_0 + i_1 x_4 + i_2 x_6}{1 + j_0 x_4 + j_1 x_6} - \beta_1 x_5$$

$$\dot{x}_6 = \frac{p_0 + p_1 x_1 + p_2 x_5}{1 + q_0 x_1 + q_1 x_4 + q_2 x_6} - \beta_6 x_6$$

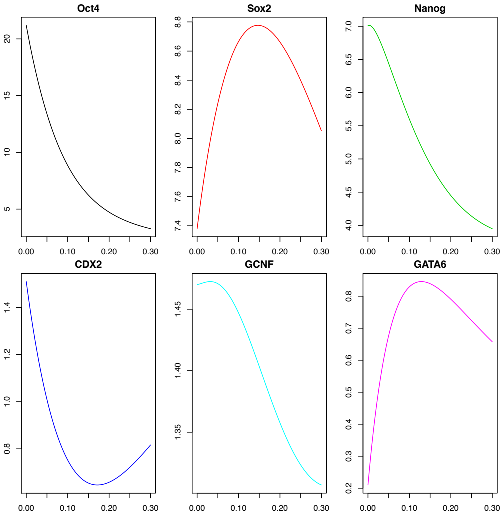
(Chickarmane, 2008)

Formal Problem Statement

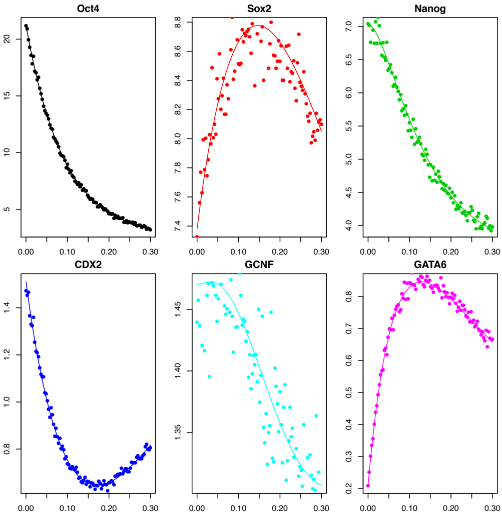
- The network to be reconstructed is the graph $\mathcal{G} = (V, \mathcal{E})$ with nodes $V = \{v_i, i = 1, ..., d\}$ corresponding to system components x_i and edges $\mathcal{E} = \bigcup E_i$.
- There is an edge $j \to i$ if $f_i(x)$ depends on x_j.
- Formalize this using partial derivatives,

$$E_i = \left\{ j = 1, ..., d : \frac{\partial f_i}{\partial x_j} \neq 0 \right\}.$$

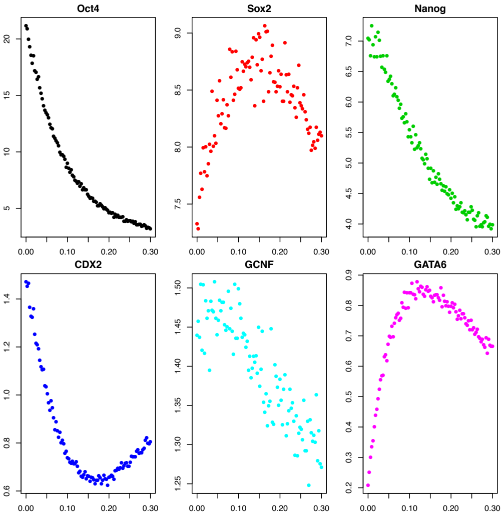
Trajectories



Trajectories



Trajectories



Formal Problem Statement

- Given noisy observations of the trajectories,

$$Y_k^r = x^r(t_k) + \epsilon_k^r, \quad \{t_k\} \subset [0,1]^n, r = 1, ..., R,$$

our goal is to estimate the edge set, $\mathcal{E}$.

- This can be viewed as a model selection problem where the goal is to estimate the nonzero elements in the Jacobian,

$$[J(f)]_{ij} = \frac{\partial f_i}{\partial x_j}.$$

Our Approach

- We do not assume knowledge of the functional form of f but instead estimate it using a nonparametric additive model,

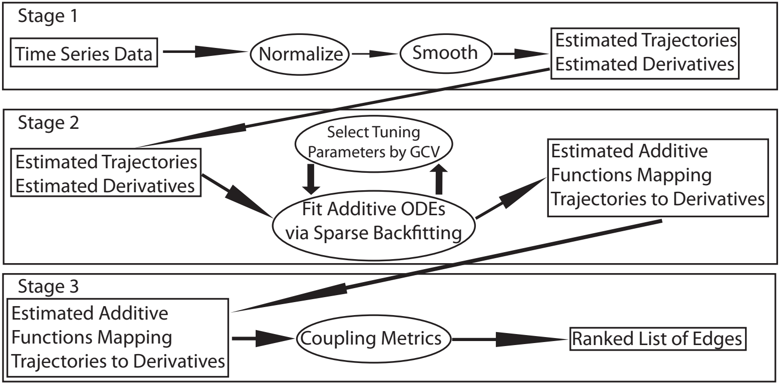
$$f = (f_1, \dots, f_d)',$$

$$f_i(x) = \alpha_i + \sum_{j=1}^{d} f_{ij}(x_j).$$

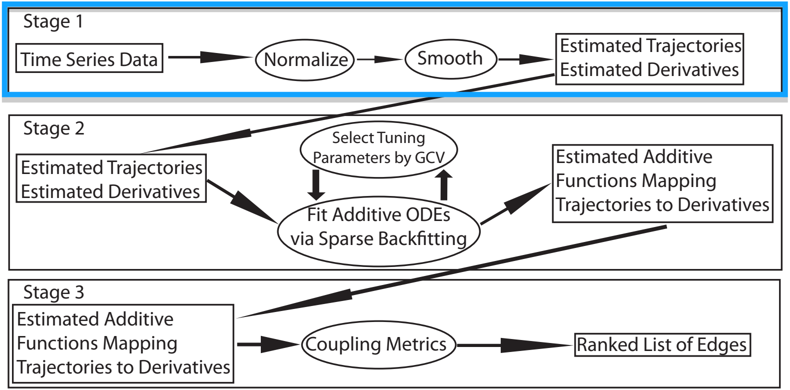
- Smoothness conditions $f_{ij} \in \mathcal{C}^2$ with $\int [\ddot{f}_{ij}(z)]^2 dz < \infty$.
- For identifiability the component functions have mean zero,

$$\int f_{ij}(x)dx = 0.$$

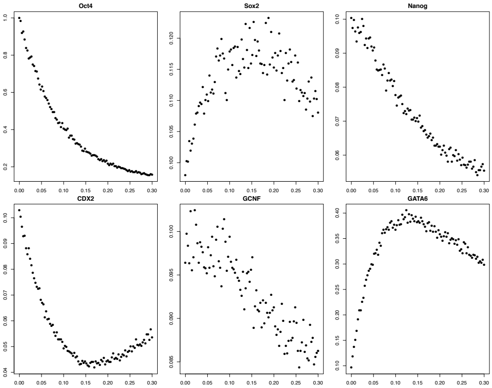
Workflow


Stage 1
Time Series Data
Normalize
Smooth
Estimated Trajectories
Estimated Derivatives
Stage 2
Estimated Trajectories
Estimated Derivatives
Select Tuning
Parameters by GCV
Fit Additive ODEs
via Sparse Backfitting
Estimated Additive
Functions Mapping
Trajectories to Derivatives
Stage 3
Estimated Additive
Functions Mapping
Trajectories to Derivatives
Coupling Metrics
Ranked List of Edges

Workflow


Stage 1
Time Series Data
Normalize
Smooth
Estimated Trajectories
Estimated Derivatives
Stage 2
Estimated Trajectories
Estimated Derivatives
Select Tuning
Parameters by GCV
Fit Additive ODEs
via Sparse Backfitting
Estimated Additive
Functions Mapping
Trajectories to Derivatives
Stage 3
Estimated Additive
Functions Mapping
Trajectories to Derivatives
Coupling Metrics
Ranked List of Edges

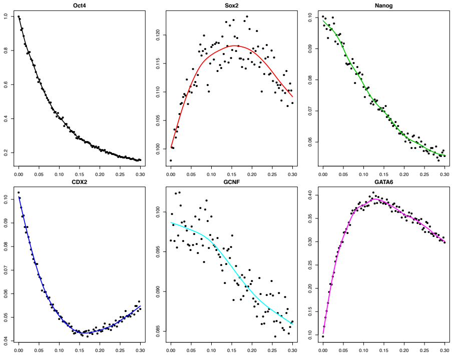
Normalize and Smooth



- Data are rescaled so that each component has maximum observation 1:

$$\tilde{Y}^r_{ik} = Y^r_{ik}/M_i \quad \text{with } M_i = \max_{k,r} Y^r_{ik}.$$

Normalize and Smooth

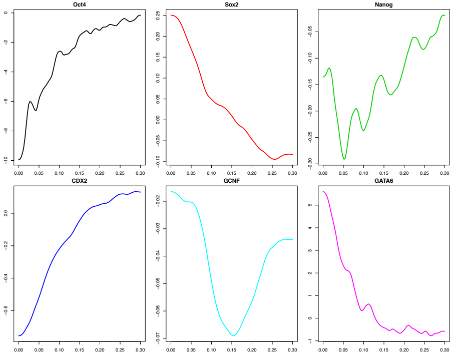


- Trajectories are estimated using smoothing splines,

$$\hat{x}_i^r = \arg \min_{x \in W_2^2[0,1]} \sum_{k=1}^{n} [\tilde{Y}_{ik}^r - x(t_k)]^2 + \lambda_0 \int_0^1 [\ddot{x}(t)]^2 dt.$$

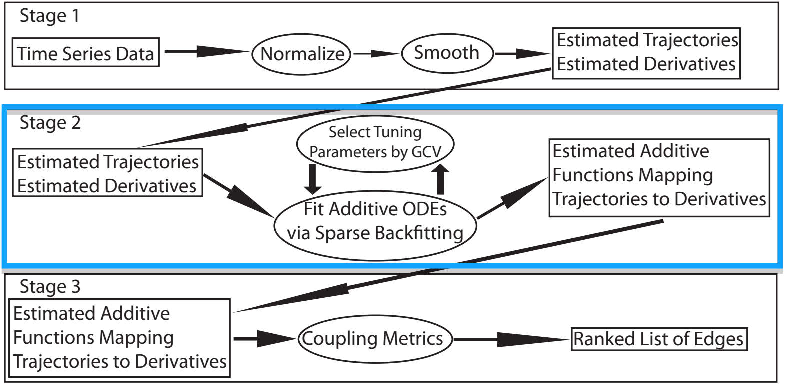
- Solution is $\hat{x}_i^r(t) = \gamma_i^r b(t)$.

Normalize and Smooth



- Estimate the derivatives using the derivative of the smoothing spline, $\hat{\dot{x}} = \gamma_i^r \dot{b}(t)$.

Workflow


Stage 1
Time Series Data
Normalize
Smooth
Estimated Trajectories
Estimated Derivatives
Stage 2
Estimated Trajectories
Estimated Derivatives
Select Tuning
Parameters by GCV
Fit Additive ODEs
via Sparse Backfitting
Estimated Additive
Functions Mapping
Trajectories to Derivatives
Stage 3
Estimated Additive
Functions Mapping
Trajectories to Derivatives
Coupling Metrics
Ranked List of Edges

Estimate an Additive ODE

- Our M-estimators are defined by the criterion,

$$\hat{M}_{n,r}(f_i) = \int_0^1 \left[\hat{\dot{x}}_i^r(t) - \sum_{j=1}^{d} f_{ij}(\hat{x}_j^r(t))\right]^2 w(t)dt + J(f_i; \lambda_1, \lambda_2)$$

- The penalty enforces both smoothness and sparsity,

$$J(f_i; \lambda_1, \lambda_2) := \lambda_1 \sum_{j=1}^{d} \int [\ddot{f}_{ij}(x)]^2 dx + \lambda_2 \sum_{j=1}^{d} \sqrt{\int [f_{ij}(x)]^2 dx}.$$

- The estimators are,

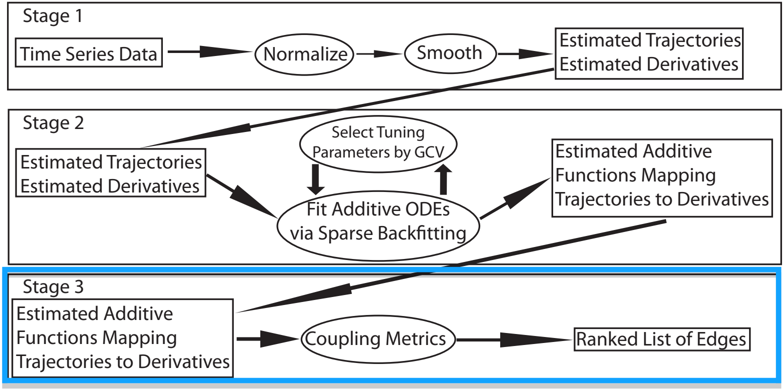
$$\hat{f}_i = \arg\min_{f_i \in \mathcal{D}} R^{-1} \sum_{r=1}^{R} \hat{M}_{n,r}(f_i).$$

- The estimator combines ideas from (Gugushvili, 2012) and (Ravikumar, 2009).

Algorithm

- The estimator is found using a modified version of the sparse-backfitting algorithm from (Ravikumar, 2009).
- Iteratively solves univariate smoothing spline problems and applies a soft-threshold.
- Each univariate smoother corresponds to a component trajectory.
- Procedure is highly parallelizable and allows for a number of numeric efficiencies.

Workflow


Stage 1
Time Series Data
Normalize
Smooth
Estimated Trajectories
Estimated Derivatives
Stage 2
Estimated Trajectories
Estimated Derivatives
Select Tuning
Parameters by GCV
Fit Additive ODEs
via Sparse Backfitting
Estimated Additive
Functions Mapping
Trajectories to Derivatives
Stage 3
Estimated Additive
Functions Mapping
Trajectories to Derivatives
Coupling Metrics
Ranked List of Edges

Coupling Metrics

- Due to the additive structure,

$$\frac{\partial f_i}{\partial x_j} = 0 \iff f_{ij} \equiv 0.$$

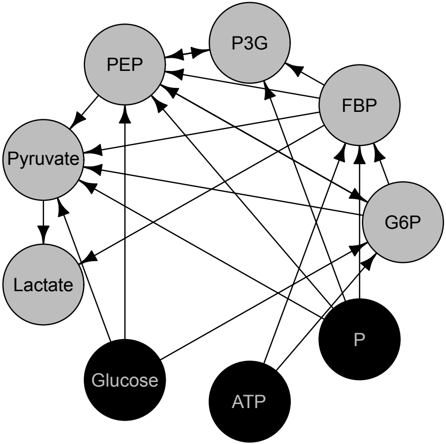
- To measure the strength of potential relationship $v_j \to v_i$ we use the coupling metric,

$$\rho_{ij} := \sqrt{\frac{\int_{\mathcal{R}_j} [\hat{f}_{ij}(z)]^2 dz}{|\mathcal{R}_j|}},$$

  with $\mathcal{R}_j$ the observed range of x_j and $|\mathcal{R}_j|$ its length.

- The ρ_{ij} are used to rank potential edges.

Glycolytic Pathway in Lactocaccus Lactis



- (Voit, 2006)
- Small network with dense edge set so fix $\lambda_2 = 0$ in advance.

Setup

- Six experimental runs over-expressing each component in turn,

$$\begin{cases} x_i^r(0) = x_{0i}, & i \neq r \\ x_i^r(0) = Mx_{0i}, & i = r. \end{cases}$$

- The trajectories were sampled at $n = 100$ times with noise added to simulate measurement error,

$$Y_k^r = x^r(t_k) + \epsilon_{rk}, \quad \epsilon_{ki}^r \overset{indp.}{\sim} N(0, [\sigma x_i^r(t_k)]^2).$$

Area under the precision-recall curve.

	$\sigma = .02$	$\sigma = .05$
M=10, Additive ODE	**.92** (.918, .920)	**.91** (.909, .912)
M=10, Linear ODE	.84 (.840, .841)	.83 (.832, .835)
M=10, Linear ODE + Lasso	.65 (.650, .657)	.67 (.669, .677)
M=10, Inferelator 1.0	.75 (.741, .750)	.74 (.734, .741)
M=5, Additive ODE	**.88** (.881, .883)	**.86** (.859, .862)
M=5, Linear ODE	.80 (.802, .804)	.78 (.776, .781)
M=5, Linear ODE + Lasso	.71 (.710, .715)	.73 (.723, .729)
M=5, Inferelator 1.0	.78 (.778, .787)	.77 (.764, .772)
M=2, Additive ODE	.55 (.549, .553)	.49 (.490, .498)
M=2, Linear ODE	.57 (.567, .569)	.57 (.567, .572)
M=2, Linear ODE + Lasso	.56 (.556, .559)	**.61** (.605, .612)
M=2, Inferelator 1.0	**.62** (.618, .624)	.60 (.592, .599)

Area under the ROC curve

	$\sigma = .02$	$\sigma = .05$
M=10, Additive ODE	**.91** (.904, .906)	**.90** (.895, .897)
M=10, Linear ODE	.83 (.826, .828)	.82 (.815, .820)
M=10, Linear ODE + Lasso	.65 (.650, .657)	.67 (.669, .677)
M=10, Inferelator 1.0	.75 (.744, .753)	.74 (.733, .742)
M=5, Additive ODE	**.87** (.871, .874)	**.85** (.852, .856)
M=5, Linear ODE	.78 (.781, .783)	.73 (.726, .731)
M=5, Linear ODE + Lasso	.71 (.710, .715)	.73 (.723, .729)
M=5, Inferelator 1.0	.77 (.764, .774)	.76 (.751, .759)
M=2, Additive ODE	**.66** (.663, .666)	.59 (.584, .591)
M=2, Linear ODE	.57 (.572, .574)	.54 (.537, .542)
M=2, Linear ODE + Lasso	.56 (.556, .559)	**.61** (.605, .612)
M=2, Inferelator 1.0	.61 (.612, .618)	.59 (.586, .597)

DREAM

- Dialogue on Reverse Engineering and Assessment Methodologies (DREAM) competitions were set up to assess network reconstruction and related methods.
- (Marbach et al 2009, 2010, 2012; Prill et al 2010)
- Data generated from realistic, thermodynamics-based *in silico* models of gene regulation.
- DREAM 3 data - knockouts, knockdowns, and multifactorial time series (4 and 46 series with $n = 21$ time points)
- We used knockouts to restrict the search space before applying additive ODEs.

Results on DREAM 3 10-Node competition data

		E1	E2	Y1	Y2	Y3
PR	Team 256	.396	.258	.258	.481	.434
	Team 304	.193	.377	.468	.332	.388
	Team 315	.710	.713	.897	.541	.627
	Additive ODEs	.875	.632	.558	.491	.510
ROC	Team 256	.720	.622	.591	.591	.625
	Team 304	.697	.791	.909	.554	.658
	Team 315	.928	.912	.949	.747	.714
	Additive ODEs	.976	.885	.906	.673	.654

Results on DREAM 3 100-Node competition data

		E1	E2	Y1	Y2	Y3
PR	Team 304	.132	.154	.159	.179	.161
	Team 315	.694	.806	.493	.469	.433
	Additive ODEs	.623	.841	.466	.424	.396
ROC	Team 304	.835	.879	.839	.738	.667
	Team 315	.948	.960	.915	.856	.783
	Additive ODEs	.867	.953	.820	.787	.734

Conclusions

- We show how nonparametric additive ODE models can be used for *de novo* network reconstruction.
- Moving from linear to additive ODEs may lead to improvements when the signal is sufficiently strong.
- Performance is comparable to top-performers on gold-standard competition data and outperforms other approaches relying primarily on time-series.

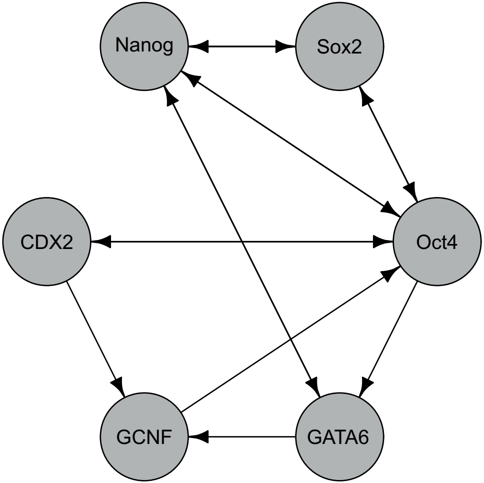
Thank You!

Questions?

For further details see: Henderson J, Michailidis G (2014) Network Reconstruction using Nonparametric Additive ODE Models. PLoS One (Forthcoming)

Send comments or additional questions to jbhender@umich.edu

Mouse Embryonic Stem Cells



- (Chickarmane, 2008)

Area under the precision-recall curve for the mouse system

	$\sigma = .02$	$\sigma = .05$
M=10, Additive ODE	**.98** (.980, .981)	**.98** (.977, .978)
M=10, Linear ODE	.96 (.963, .963)	.96 (.953, .957)
M=10, Linear ODE + Lasso	.75 (.744, .746)	.74 (.736, .741)
M=10, Inferelator 1.0	.66 (.655, .668)	.62 (.615, .629)
M=5, Additive ODE	**.98** (.984, .985)	**.98** (.979, .981)
M=5, Linear ODE	.97 (.969, .970)	.96 (.963, .965)
M=5, Linear ODE + Lasso	.75 (.751, .753)	.74 (.740, .745)
M=5, Inferelator 1.0	.70 (.696, .708)	.65 (.641, .656)
M=2, Additive ODE	**.98** (.977, .979)	.94 (.935, .941)
M=2, Linear ODE	**.98** (.976, .978)	**.96** (.953, .958)
M=2, Linear ODE + Lasso	.76 (.758, .762)	.74 (.741, .748)
M=2, Inferelator 1.0	.70 (.700, .707)	.61 (.601, .614)

Area under the ROC curve for the mouse system.

	$\sigma = .02$	$\sigma = .05$
M=10, Additive ODE	**.98** (.979, .980)	**.98** (.974, .976)
M=10, Linear ODE	.94 (.936, .938)	.93 (.926, .930)
M=10, Linear ODE + Lasso	.75 (.744, .746)	.74 (.736, .741)
M=10, Inferelator 1.0	.60 (.598, .611)	.57 (.567, .579)
M=5, Additive ODE	**.98** (.982, .983)	**.98** (.975, .977)
M=5, Linear ODE	.96 (.956, .958)	.95 (.946, .949)
M=5, Linear ODE + Lasso	.75 (.751, .753)	.74 (.740, .745)
M=5, Inferelator 1.0	.65 (.644, .655)	.60 (.588, .602)
M=2, Additive ODE	**.97** (.969, .972)	.93 (.925, .932)
M=2, Linear ODE	**.97** (.968, .971)	**.95** (.943, .949)
M=2, Linear ODE + Lasso	.76 (.758, .762)	.74 (.741, .748)
M=2, Inferelator 1.0	.66 (.658, .665)	.58 (.577, .589)