Identification of gene regulatory network from gene expression time-course data

1 Introduction and Problem Formulation

Gene regulatory networks are fundamental to understanding cellular behavior and biological processes. These networks describe how genes interact with and regulate each other, forming complex systems that control cellular functions. However, inferring these networks from experimental data remains a significant challenge in systems biology. This report addresses the problem of reverse engineering a gene regulatory network from time-series gene expression data using computational methods. We focus on a synthetic gene regulatory network in yeast, originally constructed and studied by Cantone et al. (2009). This network serves as an ideal test case for network inference methods because it is relatively small and well-defined, consisting of only 5 genes, it operates in isolation from the rest of the yeast genome and the true network structure is known, allowing for quantitative evaluation of inference accuracy.

Our objective is to infer both the structure and the nature of interactions (activation or repression) between these genes using only the time-series expression data. This represents a challenging inverse problem as the temporal nature of the data introduces dependencies between measurements, gene regulatory relationships are often non-linear, the number of possible network configurations grows exponentially with the number of genes and biological data typically contains noise and measurement uncertainties. To tackle this problem, we developed a computational approach based on ordinary differential equations (ODEs) that models the gene expression dynamics. Our method incorporates a linear ODE framework focusing on relative rates of change, structural constraints to prevent biologically implausible connections, parameter optimization with regularization to promote network sparsity and a threshold-based approach for determining significant interactions.

2 Methods

2.1 Experimental Data

The data used in this study comes from a "switch-off" time-series experiment conducted by Cantone et al. (2009). In this experiment, the activity of galactose in the system was decreased at the start of the measurements, triggering changes in the gene regulatory network. The experiment tracked the expression levels of five genes: SWI5, CBF1, GAL4, GAL80, and ASH1. The gene expression measurements were taken at regular 10-minute intervals over a total duration of 190 minutes, resulting in 20 time points per gene. Expression levels are reported in arbitrary units, with values typically ranging between 0.001 and 0.1.

2.2 Data Preprocessing

Gene expression data typically exhibits exponential behavior and spans multiple orders of magnitude. To handle this, we apply a logarithmic transformation (as discussed in the lecture on parameter optimization) with a small offset to handle near-zero values:

$$\boldsymbol{X} = \log(\boldsymbol{X} + \epsilon), \qquad \epsilon > 0.$$

After log transformation, we standardize the data to ensure all genes are on comparable scales:

$$\boldsymbol{X} = (\boldsymbol{X} - \boldsymbol{\mu}) \oslash \boldsymbol{\sigma}$$

where $\boldsymbol{\mu}, \boldsymbol{\sigma} \in \mathbb{R}^5$ denote the mean and standard deviation respectively and $\oslash$ denotes element-wise division. The scaling has been implemented using scikit-learn's StandardScaler [1], which provides robust and efficient computation of the standardization parameters

2.3 Network Inference Mode

Our approach to inferring the gene regulatory network is based on a linear ordinary differential equation (ODE) model that captures the dynamics of gene expression changes. The model is designed to identify both the structure of the network and the nature of interactions (activation or repression) between genes.

ODE Model Formulation

For a system of 5 genes, we model the rate of change in expression of each gene as:

$$\frac{dx_i}{dt} = \sum_{j \neq i} \theta_{ij} \cdot \text{sign}(x_j) \cdot |x_j| - d_i \cdot x_i \tag{1}$$

where $i, j \in \{1, ..., 5\}$, $x_i \in \mathbb{R}$, $\theta_{ij} \in \mathbb{R}$ and $d_i \in [0.01, 1.0]$. x_i is the expression level of gene i, θ_{ij} represents how gene j regulates the expression rate of gene i, and d_i is the decay rate for gene i. The term $\text{sign}(x_j)|x_j|$ ensures that:

- When $\theta_{ij} > 0$, gene j acts as a repressor of gene i (changes in opposite directions)
- When $\theta_{ij} < 0$, gene j acts as an activator of gene i (changes in same direction)

Parameter Constraints and Regularization

To ensure biological plausibility and prevent overfitting, we implement several structural constraints:

- No self-regulation: $\theta_{ii} = 0$ for all i
- No bidirectional regulation: if $|\theta_{ij}| > 0$, then $\theta_{ji} = 0$
- Positive decay rates: $0.01 \leq d_i \leq 1.0$,

as well as regularization:

- L1 regularization on interaction parameters to promote sparsity:

$$\text{L1\_penalty} = \lambda_1 \cdot \sum |w_{ij}|$$

- Quadratic regularization on decay rates:

$$\text{decay\_penalty} = \lambda_2 \cdot \sum (d_i - d_0)^2$$

  where $\lambda_1 = 0.1$, $\lambda_2 = 0.01$ and $d_0 = 0.1$ were chosen through exploratory testing to provide a balance between model fit and regularization effects..

Parameter Optimization

The model parameters are optimized by minimizing a loss function that combines prediction error with regularization terms:

$$\mathcal{L} = \sum (\hat{x} - x)^2 + \text{L1\_penalty} + \text{decay\_penalty} \tag{2}$$

where $\hat{x}$ is the ODE solution calculated by sklean's odeint function [1].

The optimization of our model parameters is performed using the Limited-memory BFGS with Bounds (L-BFGS-B) algorithm [2]. The algorithm efficiently handles our high-dimensional parameter space while respecting parameter bounds through these key steps:

1. Compute gradient $\nabla \mathcal{L}(\boldsymbol{\theta}_k)$ at the current iteration k
2. Construct a low-rank approximation $\mathbf{B}_k$ of the inverse Hessian using m previous iterations

3. Determine the search direction $\mathbf{p}_k = -\mathbf{B}_k \nabla \mathcal{L}(\boldsymbol{\theta}_k)$
4. Compute step size α_k using strong Wolfe conditions while enforcing parameter bounds
5. Update parameters: $\boldsymbol{\theta}_{k+1} = P[\boldsymbol{\theta}_k + \alpha_k \mathbf{p}_k]$

where $P[\cdot]$ denotes projection onto the feasible region defined by our bounds. This approach consistently outperformed other tested optimizers like SLSQP [3], TNC [4], and Conjugate Gradient [5] for our specific problem.

2.4 Network Structure Inference

Let $\theta \in \mathbb{R}^{30}$ be our parameter vector, where the first 25 elements form the interaction matrix $\Theta \in \mathbb{R}^{5\times5}$ and the last 5 elements represent decay rates. For each element $\theta_{ij} \in \Theta$, we determine both the presence and type of regulation of gene i by gene j using a threshold approach. For a given threshold $\tau > 0$, we establish:

- An edge exists from gene j to gene i if $|\theta_{ij}| > \tau$
- The edge represents repression if $\theta_{ij} > 0$
- The edge represents activation if $\theta_{ij} < 0$

This interpretation directly matches our ODE formulation, where θ_{ij} represents how gene j influences the expression rate of gene i.

2.5 Performance Evaluation

To assess the accuracy of our network inference method, we evaluate how well the predicted network matches the known true network structure from Cantone et al. (2009). The evaluation considers both the presence of connections between genes and their interaction types (activation or repression). We evaluate performance using standard binary classification metrics. True Positives (TP) are correctly identified interactions with correct type. False Positives (FP) are predicted interactions that don't exist in the true network, or have incorrect type. False Negatives (FN) are true interactions that were missed by our method.

From these, we compute:

- Precision $= \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$: Fraction of predicted interactions that are correct
- Recall $= \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$: Fraction of true interactions that were successfully identified
- F1 Score $= 2(\mathrm{Precision} \cdot \mathrm{Recall})/(\mathrm{Precision} + \mathrm{Recall})$: Harmonic mean of precision and recall, providing a balanced measure of performance

To understand how performance varies with different thresholds, we generate a Receiver Operating Characteristic (ROC) curve by plotting the True Positive Rate (TPR = Recall) against the False Positive Rate ($\mathrm{FPR} = \mathrm{FP}/(\mathrm{TN}+\mathrm{FP})$) across a range of threshold values. This allows us to analyze the trade-off between detecting true interactions and avoiding false predictions.

3 Experiments and Results

3.1 Experimental Setup

We implemented the network inference pipeline in Python using scientific computing libraries including NumPy, SciPy, and scikit-learn. The optimization process was initialized with zero weights for interaction parameters and 0.1 for decay rates. We sampled 50 threshold values linearly spaced between 0.05 and 2.0 to evaluate the detection of network interactions. The time points for ODE integration matched the experimental data collection intervals, ranging from 0 to 190 minutes in 10-minute steps.

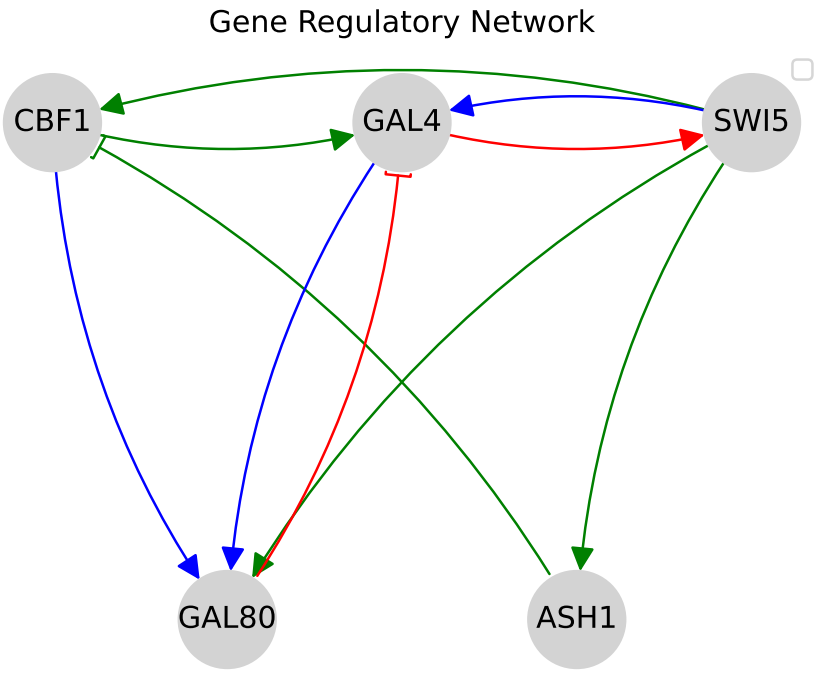

Figure 1: Reconstructed Gene Network with interactions types. Green edges represent correct interactions, red edges represent missing interactions and blue edges are predicted interactions that do not correspond to ground-truth interactions.

Metric	Structure-only	Full Inference
True Positives	5	5
False Positives	3	3
False Negatives	2	2
Precision	0.625	0.625
Recall	0.714	0.714
F1 Score	0.667	0.667

Table 1: Performance metrics at optimal threshold ($\tau = 0.09$)

3.2 Results

The optimization procedure converged successfully, producing an interaction matrix that captures potential regulatory relationships between the five genes. Performance evaluation was conducted both for network structure alone (ignoring interaction types) and for the full problem including interaction types.

Network Reconstruction

Figure 1 shows the reconstructed gene regulatory network at the optimal threshold value of 0.09. The network visualization distinguishes between correctly identified interactions (green edges), false positive predictions (blue edges), and missed true interactions (red edges). The direction of edges indicates regulatory relationships, while edge types represent activation or repression relationships.

Performance metrics for both structure-only and full network inference are presented in Table 1. Both approaches achieved identical performance, with 5 true positives, 3 false positives, and 2 false negatives, resulting in a precision of 0.625 and recall of 0.714.

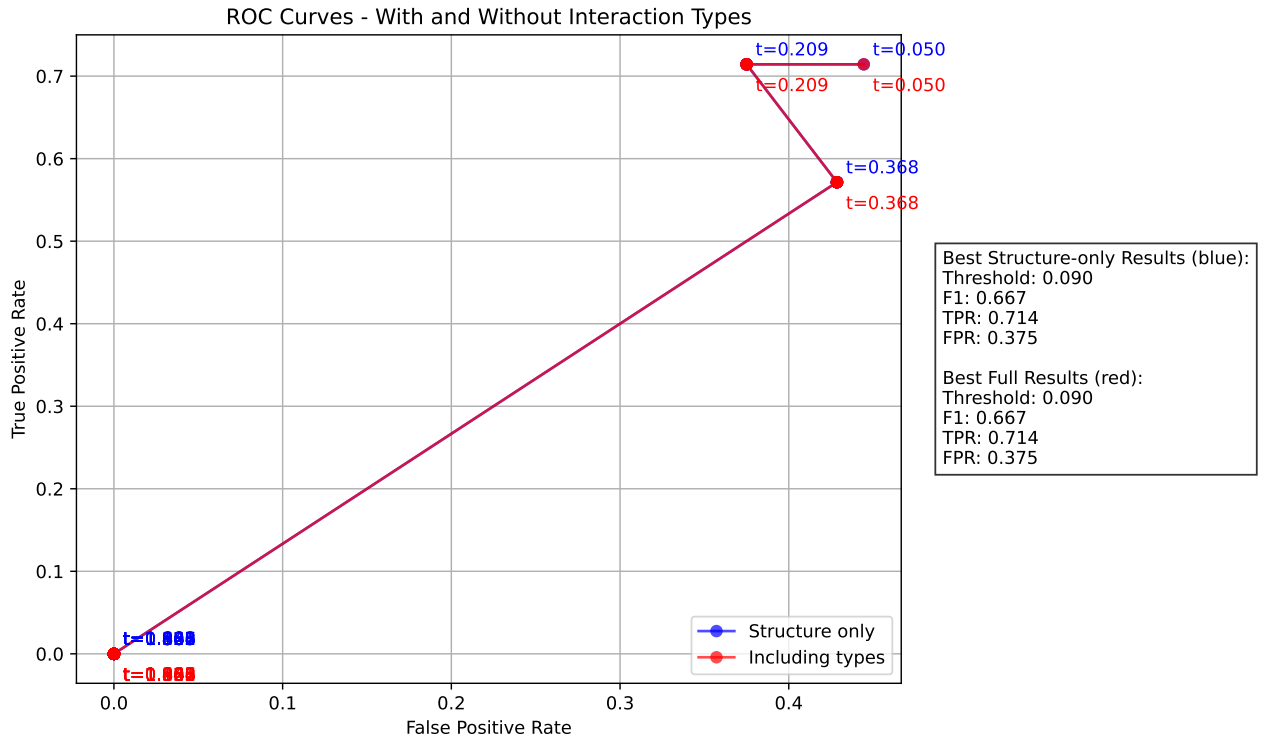

Figure 2: ROC Curves with and without interaction types.

Threshold Analysis

The relationship between detection threshold and inference accuracy is illustrated through ROC curves in Figure 2. The curves for both structure-only and full inference displayed identical characteristics, with three notable operating points:

- t = 0.050: Maximum sensitivity point
- t = 0.209: Intermediate performance point
- t = 0.368: High specificity point

The curves reached a maximum True Positive Rate of approximately 0.71 at a False Positive Rate of 0.375, corresponding to the optimal threshold of 0.090. The identical trajectories of both curves indicate that the method's ability to detect interaction types matched its ability to detect the presence of interactions.

4 Discussion

Our ODE-based approach achieved moderate success in reconstructing the gene regulatory network, with an F1 score of 0.667 indicating a balanced trade-off between precision and recall. However, several interesting patterns and limitations emerged from our analysis that warrant further discussion.

A key observation from Figure 1 is that many of the inference errors appear to be related to directionality rather than the existence of connections. For instance, while the model correctly identified several interactions between CBF1, GAL4, and GAL80, it sometimes reversed the direction of regulation. This suggests that determining causality from time-series data remains challenging, even when using differential equations that explicitly model temporal dynamics. The difficulty may arise from the complex feedback loops present in the network and the relatively sparse temporal sampling (10-minute intervals) which could mask the true sequence of regulatory events.

The identical performance between structure-only and full inference (including interaction types) is noteworthy. This suggests that when our model correctly identifies an interaction, it also correctly determines whether it is activating or repressing. This is somewhat surprising given the additional complexity of determining interaction types and might be attributed to our ODE formulation that directly encodes the nature of interactions through parameter signs.

The ROC analysis reveals that no threshold value achieves both high sensitivity and specificity, suggesting inherent uncertainty in the inferred interaction strengths. This could be improved by incorporating prior biological knowledge or additional data types such as protein-protein interactions or transcription factor binding sites.

Future improvements could focus on:

- Incorporating nonlinear regulatory functions in the ODE model
- Developing more sophisticated methods for determining interaction directionality
- Relaxing structural constraints while maintaining biological plausibility
- Using multiple perturbation experiments to better identify causal relationships

5 Conclusion

We presented an ODE-based approach for inferring gene regulatory networks from time-series expression data. Applied to a synthetic five-gene network in yeast, our method achieved an F1 score of 0.667, successfully reconstructing several key regulatory relationships. The approach demonstrated particular strength in identifying interaction types, correctly classifying connections as activating or repressing when detected. However, the challenges in determining interaction directionality highlight limitations of inference from time-series data alone. This work contributes to the field by quantifying both the capabilities and limitations of ODE-based network inference approaches. While effective for small, well-controlled systems, the results suggest that future advances might benefit from incorporating additional data types or prior biological knowledge, particularly for addressing causality in larger, more complex regulatory networks.

References

[1] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[2] Richard H. Byrd et al. "A Limited Memory Algorithm for Bound Constrained Optimization". In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208. DOI: 10.1137/0916069.

[3] Dieter Kraft. "A software package for sequential quadratic programming". In: *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt* (1988).

[4] Stephen G Nash. "Newton-type minimization via the Lanczos method". In: *SIAM Journal on Numerical Analysis* 21.4 (1984), pp. 770–788.

[5] Magnus Rudolph Hestenes, Eduard Stiefel, et al. *Methods of conjugate gradients for solving linear systems*. Vol. 49. 1. NBS Washington, DC, 1952.

AI Disclaimer

This work was conceptualized and executed by a human. The textual formulation and presentation of ideas in this report have been enhanced with the assistance of a large language model. All core scientific content, analyses, and conclusions remain the original work of the human author.