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

Katariina Yliräisänen (100839171), Laura Leino (100768721)

December 2025

1 Introduction

The aim of this project was to infer a gene regulatory network from time-series gene expression measurements and evaluate how well the network matches a known reference network. The regulatory network is represented as a set of directed edges from transcription factors (TFs) to target genes. The analysis is based on a dataset from Cantone et. al. (2009) [1], who reported a small synthetically constructed transcriptional network in yeast consisting of five TF genes regulating each other. Because endogenous yeast genes are assumed to have a negligible effect, the system can be treated as an isolated five-gene network.

The dependencies between the genes are modeled using a **Dynamic Bayesian Network (DBN)** with a four-step lag. In this setup, potential regulatory influences are represented as directed edges from gene expression at time t to gene expression at time t + 4. The network structure is then learned using hill-climbing search under two alternative discrete scoring criteria: the K2 score and the BIC-d score. Two scoring criteria were used because K2 and BIC-d highlight different trade-offs between model fit and complexity. Comparing them helps to evaluate how sensitive the inferred edges are to the choice of scoring method. To asses how stable the inferred interactions are, bootstrap resampling is applied and each edge is assigned a confidence score based on how often it is selected across bootstrap runs. Performance is then evaluated against the reference standard using ROC curves and AUC.

2 Dataset Description and Data Preprocessing

The dataset is based on data published by Cantone et al. (2009) on synthetically constructed gene regulatory network in yeast [1]. It analyzed consists of expression profiles for five genes (*SWI5, CBF1, GAL4, GAL80,* AND *ASH1*) measured at 10-minute intervals from 0 to 190 minutes. The dataset contains 20 rows and 6 columns, with five gene expression variables and one time variable. To obtain reliable results, the time column was removed at the beginning of the data preprocessing.

The structure-learning methods applied in this project operate on discrete variables, so the expression values are transformed into binary states using quantile binning (q = 2). A two-group discretization was chosen because the dataset is relatively small. Using more groups could have led to overfitting and made the model less reliable. The first group represented low gene expression, meaning that the gene was weakly active at that time point. The second group represented high gene expression, meaning that the gene was active during that time point. This type of division helps to reduce noise in the data while keeping the most important information. For each gene g, a threshold is defined as the empirical median of its measurements, and the discretized value is

$$z_{g,t} = \begin{cases} 0, & x_{g,t} \leq \text{median}(x_g) \\ 1, & x_{g,t} > \text{median}(x_g) \end{cases} \qquad z_{g,t} \in \{0, 1\}.$$

To model one-step temporal dependencies, the time series is transformed into a transition dataset. Each transition samples is created from two consecutive time point by concatenating the variables at time t (slice 0) with the variables at time $t + \Delta t$ (slice 1). Gene expression is relatively slow biological process. The effect of transcription factors on the expression of their target genes is usually observed after a time delay. For this reason, the gene states at time point t were used to explain the gene state at time point $t + 4$, corresponding to four measurement intervals (40 minutes).

$$Z_t = (z_{1,t}, z_{2,t}, \ldots, z_{5,t})$$

$$Z_{t+4} = (z_{1,t+4}, z_{2,t+4}, \ldots, z_{5,t+4})$$

The optimal time lag was determined by testing several different intervals and evaluating the resulting model performance. This time delay allows regulatory effects to be modeled on a more realistic time scale and reduces the risk of including biologically unlikely interactions. At the same time, the chosen interval preserves a enough observations for model, which is important given the small size of the dataset.

3 Method

3.1 Dynamic Bayesian Network

A Dynamic Bayesian Network (DBN) was chosen to infer the gene regulatory network [2]. It is well suited for modeling time-series data and for representing directed temporal dependencies between variables. Gene expression is a dynamic process, and the effects of transcription factors on their target genes are typically observed after a time delay. A DBN explicitly models such delayed dependencies between time points, making it a statistically suitable framework for gene regulatory network inference. [2]

Bayesian network is a probabilistic graph that models conditional dependencies between variables [2]. A Dynamic Bayesian Network extends the concept to time dependent system. It shows the systems's state over time. It tells how variables influence on each other over time. DBN is based on the Markov assumption, which states that the current state of the system depends only on the immediately preceding state. The earlier history is assumed to have no direct effect. [2] This means that the gene expression states at time t + 4 depends only on the expression states at time t.

$$P(Z_{t+4}|Z_t, Z_{t-4}, Z_{t-8} = P(Z_{t+4}|Z_t)$$

Temporal dependencies are modeled with a two-slice Dynamic Bayesian Network (DBN), because it provides a probabilistic framework for time-series network inference.

$$P(\mathbf{z}_{t+1} \mid \mathbf{z}_t, G, \theta) = \prod_{g=1}^{5} P\big(z_{g,t+1} \mid \mathbf{z}_{\mathrm{Pa}(g),t}; \theta_g\big)\, P(D \mid G, \theta) = \prod_{t=0}^{T-2} \prod_{g=1}^{5} P\big(z_{g,t+1} \mid \mathbf{z}_{\mathrm{Pa}(g),t}; \theta_g\big)\,.$$

This method allows both the network structure and the model parameters to be learned directly from the data. The structure learning identifies which genes are likely to regulate others. [2]

3.2 Structure learning using Hill-Climbing search

The DBN structure is learned by a greedy hill-climbing search, which aims to maximize a score [2]. The method finds a network structure G that maximizes a scoring function given the data D:

$$\hat{G} = \arg\max_{G} \mathrm{Score}(G|D)$$

Hill climbing is a greedy search algorithm. At each iteration, the algorithm evaluates local modifications (edge add/remove/reverse) and applies the change to give the largest score improvement. The algorithm stops when no further improvements are found [2]. In this study, the search was terminated when the score improvement fell below $\epsilon = 10^{-4}$.

To ensure that only biologically possible dependencies are found, the allowed edges are restricted to run from the past time slice to the future time slice. This means that the only possible dependencies are from t to t + Δt, meaning edges directed from the past time slice to the future time slice. The modeling was completed with two alternative scoring functions: the K2 score and the discrete Bayesian Information Criterion (BIC-d)[2]. Both score functions are designed for discrete Bayesian network models.

3.2.1 K2 score

The K2 score is a Bayesian scoring function that measures the posterior probability of a network structure. It assumes that the conditional probability distributions of variables follow multinomial distributions with Dirichlet priors. The K2 score for network structure G can be written as

$$\text{Score}_{\text{K2}}(G|D) = \Pi_{i=1}^{n}\Pi_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\Pi_{k=1}^{r_i}N_{ijk}!$$

The K2 score prefers network structures that explain the observed data well while using prior assumptions defined by a Dirichlet distribution. Because the K2 score does not include an explicit penalty for model complexity, it can favor more complex network structures. [2]

3.2.2 BIC-d score

The discrete Bayesian Information Criterion (BIC-d) is a scoring method. It is used to select a network structure by balancing how well the model fits and model complexity. Unlike the K2 score, BIC-d includes an explicit penalty for model complexity. This helps to prevent too complicated networks structures.

The BIC-d score for network structure G is defined as

$$\text{Score}_{\text{BIC}}(C) = \log P(D|\hat{\theta}, G) - \frac{k}{2}\log N$$

where $\log P(D|\hat{\theta}, G)$ is the log-likelihood of the data based on the estimated model parameters. k is the number of free parameters in the network and N is the number of observations. The log-likelihood measures how well the network explains the observed data. The penalty term increases as the number of parameters grows. As a result, edges are added to the network only if they significantly improve the fit to the data. BIC-d score generally favors simpler network structures than the K2 score, especially when the amount of data is limited.[2]

3.3 Bootstrap resampling and edge confidence estimation

Biological gene expression data typically contains a lot of noise. As a result, structure learning methods such as hill climbing may identify edges that do not correspond to real dependencies. This problem is especially relevant when the dataset is small. Bootstrap method is used to reduce the influence of noise and to identify more reliable dependencies in the network. The main idea of the bootstrap method is to repeat the structure learning process several times using slightly different versions of the data. This makes it possible to evaluate which edges are consistently found. If edges appear only few times they are most likely caused by the noise and therefor are excluded from the network. [2]

Multiple dataset are created by sampling the original transition samples with replacement. Each bootstrap dataset has the same size as the original dataset but contains different combinations of observations. The network for each bootstrap dataset is learned by using the same hill-climbing function. In the study process was repeated 200 times and the results were combined by counting how often each possible edge appeared in the networks. For each edge an confidence score was calculated as

$$s_e = \frac{1}{B}\sum_{b=1}^{B} I(e \in \hat{G}^b$$

where I is an indicator function that equals 1 if the edge is present in the network learned from bootstrap dataset and 0 otherwise. The confidence score s_e takes values between 0 and 1 and describes how often an edge is included in the bootstrap runs. Edge with high confidence scores are more likely to represent true connection, while edges with low scores are more likely noise. The final network is constructed by selecting edges whose confidence scores are bigger than a chosen threshold. By changing the threshold, the balance between finding true interactions and avoiding false positives can be examined. [2]

4 Performance evaluation

To evaluate the accuracy of the two inferred gene regulatory networks, the edge-level prediction performance was evaluated against a known reference network. The evaluation was done at the edge level by varying a detection threshold and reporting how many true and false TF-target edges are detected at each threshold. Because structure learning on a short and noisy time series could be unstable, edge support was quantified using bootstrap resampling ($N = 200$). Transition samples were resampled with replacement from multiple bootstrap datasets of the same size as the original. A DBN structure was learned separately for each bootstrap dataset using the hill-climbing algorithm. The results were collected by counting how often each candidate edge appeared. For each directed edge e, a confidence score $s_e \in [0, 1]$ was computed using

$$s_e = \frac{1}{B} \sum_{b=1}^{B} \mathbf{x}\Big(e \in \hat{G}_b\Big),$$

where $\hat{G}_b$ is the network learned in bootstrap run b, and x equaled 1 when the edge was found and 0 otherwise. A predicted network at threshold τ was defined as $\hat{E}(\tau) = \{\, e : s_e \geq \tau \,\}$. For each τ, the required fractions were computed by comparing $\hat{E}(\tau)$ to the reference networks edge set. The fraction of true interactions detected corresponded to the true positive rate (TPR), and the fraction of false interactions detected corresponded to the false positive rate (FPR).

For performance evaluation, a receiver operating characteristics (ROC) curve was created. The ROC curve summarizes the trade-off between correctly detected interactions and false positives by plotting the TPR against FPR at different confidence thresholds. For each τ, the pair (FPR(τ), TPR(τ)) defined one point on the ROC curve, with FRP on the x-axis and TPR on the y-axis. In addition, the area under the ROC curve (AUC) reported. AUC measures the overall ranking performance above non-reference interactions. In addition, the smallest threshold τ was identified and the amount of false interactions was reported at his threshold.

5 Results

The aim of the modeling was to identify a dynamic network that describes gene regulatory interactions. In this study, two different scoring methods were used in order to enable comparison between the approaches. Figures 1 and 2 show the Dynamic Bayesian Network model based on the two different scoring methods.

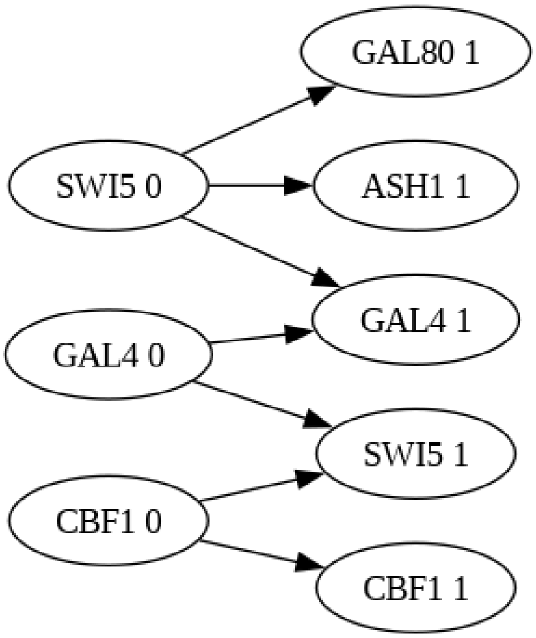

Figure 1: K2 Scoring

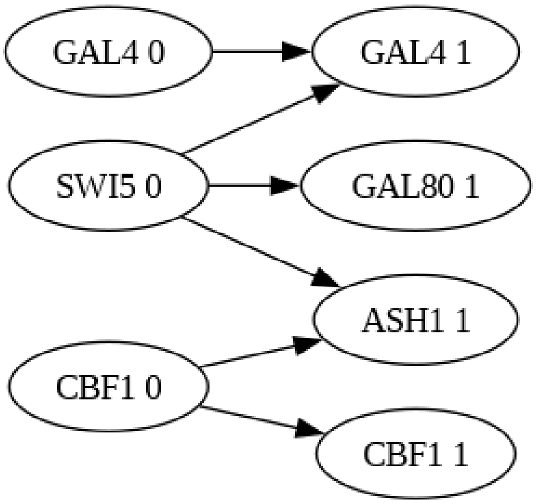

Figure 2: BIC-d Scoring

Dynamic Bayesian Network with K2 and BIC-d scoring

With K2 scoring method, the inferred network contains seven connections. In contrast with the BIC-d scoring method the network contains six connections. In both networks, the parent nodes consist of the same genes: SWI5, GAL4 and CBF1. Similarly, the child nodes in both networks are GAL80, ASH1, GAL4 and CBF1. SWI5 appears also as a child node in network with K2 method. In both networks, SWI5 forms connections with GAL80, ASH1 and GAL4, and no additional outgoing connections are observed. GAL4 forms a self-regulatory connection in both networks and in K2 scoring network it also regulates SWI5. CBF1 forms a self-regulatory connection in both networks. In addition CBF1 regulates SWI5 in the K2 scoring network and with ASH1 in BIC-d network.

Cantone et al. have previously characterized the structure of the synthetic five-gene regulatory network used in this study[1]. The results can be compared to the known network structure. The interactions of the known structure are presented in Table 1.

Table 1: Known network structure [1]

Parent node	Child node
SWI5	GAL80
SWI5	CBF1
SWI5	ASH1
GAL4	SWI5
CBF1	GAL4
ASH1	CBF1

Both methods successfully identified the interactions between SWI5 and GAL80, CBF1 and ASH1. The network using K2 scoring method also identified a connection between GAL4 and SWI5. The network using BIC-d scoring correctly detected the interaction between CBF1 and ASH1; however the direction was inferred incorrectly.

Both networks contained a small number if false-positive connections. The Network with scoring method K2 contained three false positives. The connections were from GAL4 to itself and CBF1 to SWI5 and to itself. The network with scoring method BIC-d contained two false-positives connections in addition to the incorrectly directed edge. These false positives were self-connections involving genes CBF1 and GAL4.

ROC analysis, shown in Figure 3, was used to summarize how the bootstrap confidence scores separated true TF → target interactions from non-reference interactions by comparing the detected edges to the reference network listed in Table 1. In the ROC plot, the TPR was plotted against the FPR for different confidence thresholds τ. The dashed diagonal like represents random ranking, while curves closer to the upper-left corner indicate better performance.

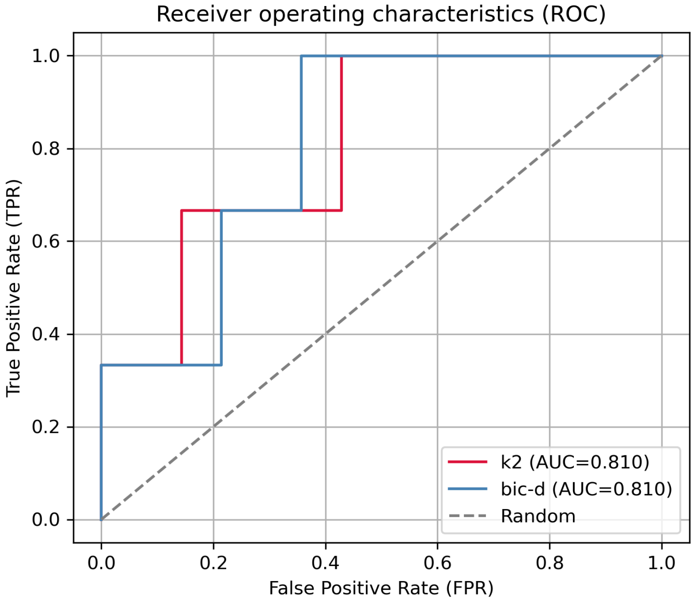

Figure 3: Receiver operating characteristics (ROC) curves for edge detection

Both ROC curves are located clearly above the random baseline for most of the range, indicating that the confidence scores provided informative ranking, with true interactions generally receiving higher scores than non-reference edges. The curves have a step-like shape, which was expected because the number of reference positives was small (six interactions), so each additional detected true edge caused a relatively large increase in TPR.

AUC measured the area under the ROC curve, and it can be interpreted as the probability that a randomly chosen true edge received a higher score than a randomly chosen false edge. Both scoring methods produced the same AUC value 0.81, as shown in Table 2, suggesting that the overall ranking quality of K2 and BIC-d was essentially identical in this experiment.

Table 2: Receiver operating characteristics (ROC) results

Scoring method	AUC	Threshold for all 6	False Positives to get threshold
K2	0.81	0.28	6.00
BIC-d	0.81	0.12	5.00

In addition to AUC, the operating point corresponding to full recall was reported, i.e. the smallest confidence threshold τ that still recovered all six reference interactions. According to Table 2, this point was reached at $\tau = 0.28$ for K2 (TP = 6, FP = 6) and $\tau = 0.12$ for BIC-d (TP = 6, FP = 5). These results suggest that bootstrap confidence scores were distributed differently across edges for the two scoring algorithms. K2 required a higher cut-off to include the last missing true interaction, whereas BIC-d reached full recovery at lower confidence level. In both methods, achieving full recall also required accepting non-reference edges in addition to the six true interactions. At this operating point, BID-d produced one fewer false positives than K2.

6 Conclusion

The aim of this work was to model gene expression and regulatory effects between genes from time-series data. A Dynamic Bayesian Network was used to identify regulatory interactions that influence gene expression levels. The modeling was performed twice using two different scoring methods, K2 and BIC-D. Both methods identified regulatory interactions involving the SWI5 gene and its effects on other genes. In addition, the K2 method identified a connection between GAL4 and SWI5. Overall, the methods were relatively successful when compared to a reference network. However, some connections were not detected, and some incorrect interactions were included in the inferred models.

The K2 method is optimistic, leading to denser networks that include multiple false positive edges. In contrast, the BIC-D method is more conservative and adds edges less frequently. As a result, false positive interactions are generally less common, but true positive interactions may also be missed more easily. Based on the ROC analysis, both methods achieved identical AUC values. The K2 method correctly predicted four true regulatory interactions, while the BIC-D method predicted three. Overall, both methods performed at a comparable level.

However, there is still much to improve in the results. All six known regulatory interactions should be identified for accurate model. The limitations of the model are likely influenced by the short and limited dataset used in the study. Model performance could be improved by extending the duration of the measurements and by collecting multiple independent time-series experiments. With larger datasets, more complex models could also be applied, as the risk of overfitting would be reduced. This would allow the detection of regulatory features and interactions that may currently remain hidden due to limited data and simplified modeling assumptions. A larger dataset would also help to recognize true signals from noise.

In conclusion, the DBN approach worked fairly well for modeling gene regulation from time-series data. Both scoring methods were able to detect some known interactions, and overall their performance was quite similar. At the same time, the inferred networks still missed some true connections and included some incorrect ones, which shows that the results are still limited. With longer and more replicated time-series datasets, the model would likely identify more of the expected regulatory interactions and thus improve the accuracy of the inferred network.

References

[1] I. Cantone et al., “A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches,” *Cell*, vol. 137, 2009. DOI: `10.1016/j.cell.2009.01.055`. [Online]. Available: `https://doi.org/10.1016/j.cell.2009.01.055`.

[2] K. P. Murphy, *Machine Learning: A Probabilistic Perspective.* MIT Press, 2012, ISBN: 9780262304320. [Online]. Available: `https://books.google.fi/books?id=RC43AgAAQBAJ`.