Discovery of Regulatory Connections in Microarray Data

Michael Egmont-Petersen, Wim de Jonge, and Arno Siebes

Institute of Information and Computing Sciences, Utrecht University,
Padualaan 14, De Uithof, Utrecht, The Netherlands
Michael@cs.uu.nl

Abstract. In this paper, we introduce a new approach for mining regulatory interactions between genes in microarray time series studies. A number of preprocessing steps transform the original continuous measurements into a discrete representation that captures salient regulatory events in the time series. The discrete representation is used to discover interactions between the genes. In particular, we introduce a new across-model sampling scheme for performing Markov Chain Monte Carlo sampling of probabilistic network classifiers. The results obtained from the microarray data are promising. Our approach can detect interactions caused both by co-regulation and by control-regulation.

1 Introduction

In bioinformatics, we are faced with an increasing amount of data that characterize the structure and function of different living organisms. Still more experimental data such as sequences (nucleotides, proteins) and gene activities (mRNA expression ratios) are generated either in the biology laboratory or in a clinical setting. The ever-expanding datasets fuel a growing demand for new datamining techniques that can help to discover possible relations between the biological entities under study and couple the different sources of data. Such datamining techniques should be able to cope with many variables that may exhibit complex dependency relations. We present a new cross-model sampling Markov Chain Monte Carlo algorithm, which we test by learning Bayesian network classifiers to predict regulatory relations between a set of predictor genes and a target gene.

Microarrays were introduced in the nineties as a means for studying in parallel the expression of all genes pertaining to a particular organism. One of the ultimate goals is to discover which genes are involved in the regulation of others, the so-called *regulatory pathways*. Microarrays measure the relative abundance of mRNA, corresponding to each known gene transcribed at a certain time t in a particular organism under study. So the prospect of microarrays is that of an aid that can help to identify functional roles of genes and eventually enrich the knowledge of the complex relations between the genotype and the phenotype of the organism under study.

Microarray time series experiments are conducted in order to study significant dynamic expression patterns. One goal of a time series experiment is to investigate which genes regulate others. It is to be expected that some genes that are controlled by the

J.-F. Boulicaut et al. (Eds.): PKDD 2004, LNAI 3202, pp. 149–160, 2004.

same transcription factor show a similar but lagged expression pattern over time, when the expression of the particular transcription factor varies. We make a distinction between *co-regulation* and *controlled regulation*. Two genes are said to be positively co-regulated when the change in relative abundance of the genes has the same first- order derivatives with respect to time. Two genes are said to be inversely co- regulated when the change in relative abundance of the genes has the opposite first-order derivatives with respect to time. Two perfectly co-regulated genes can have expression patterns with different amplitudes. One or more genes (the regulators) are said to control the expression of a particular gene (the target) when the expressions of the regulator genes directly influence the expression of the target gene.

Under conditions where particular genes are co-regulated or one or more regulator genes control the expression of a target gene, one would expect co-variation between the expressions of these genes over time. Our goal is to develop a datamining approach that can discover dynamic patterns of co-regulation and control regulation between sets of genes. Clustering techniques and correlation measures have been used extensively to identify groups of genes that are likely to be functionally related, see, e.g., Datta & Datta for an overview [1]. However, the standard clustering techniques do not take post-transcriptional and post-translational lag times into account. More importantly, in mining the vast amount of time series array data for putative control regulation relations, *lags* between expression levels of genes may contain indicative clues as to which genes code for proteins that act as regulators for others.

In this article, we present a novel datamining method for finding possible regulatory relations between small sets of genes, based on time-course microarray data, see, e.g., [2]. In the sequel, we regard the normalized (relative) expression levels of each gene as a time signal. We introduce preprocessing steps that transform such a time signal into "salient features", points in time that may disclose possible lagged interactions between genes. From this discrete representation, we train dynamic Bayesian networks to predict regulatory events of specific target genes using a novel MCMC-approach. Our new method is evaluated on microarray data obtained from the experiments by Spellman et al. [2]. The results are promising. Most of the regulatory relations found could be corroborated by literature.

2 Microarray Data

Our goal is to discover and interpret statistical relations between the relative expression of genes. For that purpose, we need to choose a suitable representation scheme for time series microarray data. Generally, each spot indicates the average relative (log) expression of mRNA corresponding to a particular gene R_i. The expression ratio of gene R_i can be seen as a continuous stochastic variable, characterized by the probability density function $p(R_i)$. Each variable R_i can either be a predictor or a target, relative to the other variables entering the model. We use t to denote the time step at which variable R_i is being measured and discretize the arraydata $R_i(t)$, $t \in \{t_0, \ldots, t_T\}$ into the following three categories: *change*, *local minimum* and *local maximum*. This differs from the approach by others [3, 4], who make a distinction between up-regulated, medium regulated and down-regulated gene expression. Our representation is different in the sense that it combines successive expression ratios $R_i(t_{v-1})$, $R_i(t_v)$ and $R(t_{v+1})$ into fea-

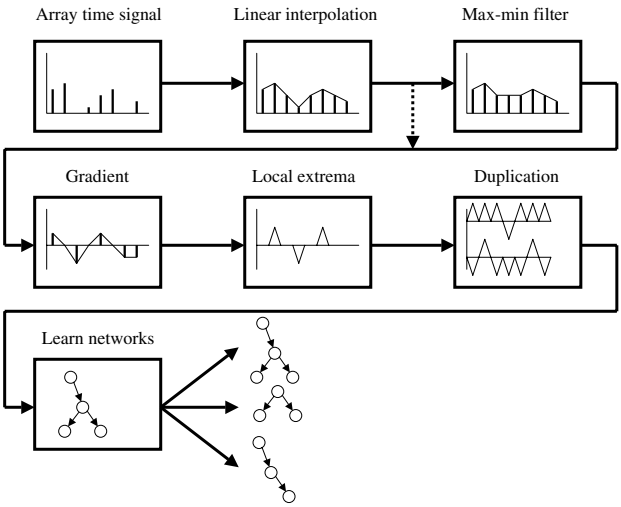

Fig. 1. In total six preprocessing steps are performed before our new datamining algorithm is applied to the dataset: 1) the log ratios of each gene are computed, 2) linear interpolation results in uniformly sampled log expression ratios, 3) the (optional) max-min filter removes transient extrema, 4) convolution with the first-order derivative of the Gaussian function results in derivatives of the expression ratios, 5) the local extrema are defined as time points at which the sign of the first-order derivative changes, 6) the selected target gene is coded as a binary variable by duplication, 7) MCMC-learning of local genetic networks.

tures that capture the local dynamics (local extrema) of the expression ratios. With our representation, relations are discovered between the most likely time points at which a gene (eventually its associated protein) is active (local maximum) and inactive (local minimum). Our approach makes it possible to establish a regulatory relation between a transcription factor with small absolute changes in expression ratio, and a target gene, because the amplitude is disregarded.

The preprocessing steps consist of 1) computation of the log-ratio per gene, 2) linear interpolation, 3) max-min filtering (optional) and 4) detection of local minima and local maxima using the derivative operator from the linear scale space. In the steps 5) and 6), the local extrema are identified and the number of observations doubled.

2.1 Interpolation

Computation of the derivatives over each gene entails the application of (linear) filters. Filtering requires that the signal be uniformly sampled over time. We use a linear nearest neighbor scheme to interpolate non-uniformly sampled time series because this scheme can never introduce new local minima or maxima. Interpolation results in a uniformly sampled time series $t,\ t \in \{1, \ldots, T\}$ of expression rations, $R_i(t)$ for gene i.

2.2 Max-Min Filter

To cope with transient changes in the first order derivative as a result of noise, we incorporate an extra (optional) preprocessing step consisting of the morphological max-

min filter [5]. An advantage of the max-min filter is that non-transient extrema in the original signal are left unaffected. The max-min filter is defined as

$$K(t) = \frac{\max\limits_{t_1 \in b(t)} \left(y = \min\limits_{t_2 \in b(t_1)} (R(t_2)) \right) + \min\limits_{t_1 \in b(t)} \left(y = \max\limits_{t_2 \in b(t_1)} (R(t_2)) \right)}{2} \quad (1)$$

When the width of the window $b(t)$ exceeds zero, small inflections become saddle points, otherwise $K(t) = R(t)$.

2.3 Regularized Differentiation

Transformation of the continuous expression ratios $K_i(t)$ into the desired discrete representation: *change*, *local minimum* and *local maximum*, requires the computation of derivatives, $\partial K_i(t)/\partial t$. We use operators from the linear scale space [6, 7] to transform differentiation into a well-posed problem [8] by means of regularization. Regularized derivatives of a discrete time series are obtained by convolution with the first-order derivative of the Gaussian function

$$g\prime(t;\mu,\sigma) = \frac{-\sqrt{2}}{2\sqrt{\pi}\sigma^3} \cdot (t-\mu) \cdot \exp\left(-\frac{(t-\mu)^2}{2\sigma^2} \right) \quad (2)$$

Convolution with $g\prime$ results in

$$H(t) = g\prime(t;0,\sigma) * K(t) = \int_{-\infty}^{\infty} g\prime(\tau;0,\sigma) \cdot K(t-\tau)\, d\tau \quad (3)$$

When the sign of $H(t)$ changes between two consecutive time steps, $H(t-\delta) < 0$ but $H(t+\delta) > 0$, this indicates a *local minimum* whereas $H(t-\delta) > 0$ but $H(t+\delta) < 0$ indicates a *local maximum*. When there is no change in sign, the time step $H(t)$ gets the label *change*.

2.4 Data Representation and Modeling

Our goal is to identify possible co-regulatory and control-regulatory relations between sets of genes. With $\mathcal{C}(R_i, R_j)$ we indicate co-regulation between the genes R_i and R_j, whereas $\mathcal{T}(R_i \rightarrow R_j)$ indicates that gene R_i controls the regulation of gene R_j. An important difference between co-regulation and controlled regulation is that co-regulation is a commutative relation, whereas controlled regulation is assumed not to be commutative. Consequently, the inclusion of lags in the time series should, in theory, make it possible to discern putative control regulations from co-regulations. The continuous valued variable $H(t)$ is discretized by the function f. This results in a discrete time series per gene, $x_{i,t} = f(H_i(t-\delta), H_i(t+\delta))$, with $X_{i,t} = x_{i,t},\ x_{i,t} \in \{\min, \text{change}, \max\}$.

We propose to model possible gene interactions using dynamic Bayesian network classifiers. In the remaining part of the paper, we use X to indicate the set of predictor genes and C the target gene. Figure 2 indicates which types of relations may be found by our approach. We use a lagged time series model in the following way. At each time

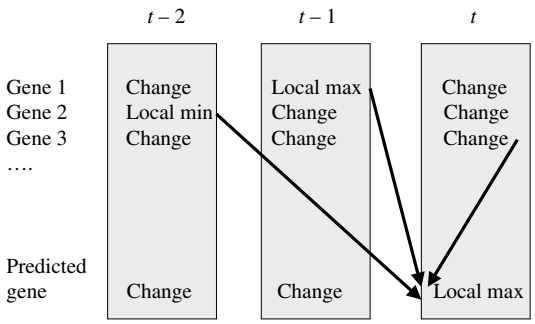

Fig. 2. Significant regulatory events of the expression of a target gene are being predicted by regulatory events pertaining to other genes earlier and at the same time as the target. Thereby, both control regulation and co-regulation with predictor genes can be modeled.

step t, the outcome of one target gene C_t, c_t, should be predicted by the outcomes of the predictor genes $x_{i,\tau}$, $\tau \in \{t-\lambda, .., t\}$ with λ, $\lambda \geq 0$, indicating the maximal lag that can be accounted for. This representation results in the following matrix of $n = r \times \lambda$ (potential) predictor variables

$$\boldsymbol{X}_t = \begin{bmatrix} x_{1,t-\lambda} & x_{1,t-\lambda+1} & \cdots & x_{1,t} \\ x_{2,t-\lambda} & \cdots & \cdots & x_{2,t} \\ \cdots & \cdots & \cdots & \cdots \\ x_{r,t-\lambda} & \cdots & \cdots & x_{r,t} \end{bmatrix} \tag{4}$$

from which the outcome of C_t is being predicted. The data $(\boldsymbol{X}_t, C_t)$, $t \in \{1, \ldots, T\}$, constitutes the basic training set. We use a datamining algorithm that performs concomitant feature and model selection in order to estimate the most likely lagged classifier model. Connections to features (possible regulator genes) that contribute to predicting the outcome of C_t are likely to be included in the model whereas genes that do not improve the predictive performance remain disconnected.

For most of the genes, a local extremum occurs much less frequently than a change. Consequently, it is likely that many correlations appear between genes of which the expression changes. To ensure that only local minima and local maxima of the target gene are being predicted, we choose to reduce the number of possible outcomes of the target variable to just two: *local minimum* and *local maximum*. To retain the three basic outcomes, we double the number of observations resulting in a final training set $\boldsymbol{D}$

- $(\boldsymbol{x}_t, c_t = max)$, becomes $\boldsymbol{d}_s = (\boldsymbol{x}_t, c_t = max)$ and $\boldsymbol{d}_{s+1} = (\boldsymbol{x}_t, c_t = max)$
- $(\boldsymbol{x}_t, c_t = min)$, becomes $\boldsymbol{d}_s = (\boldsymbol{x}_t, c_t = min)$ and $\boldsymbol{d}_{s+1} = (\boldsymbol{x}_t, c_t = min)$
- $(\boldsymbol{x}(t), c_t = change)$, is doubled into an ambiguous prediction of c_t, $\boldsymbol{d}_s = (\boldsymbol{x}_t, c_t = min)$ and $\boldsymbol{d}_{s+1} = (\boldsymbol{x}_t, c_t = max)$

with $s = 1, \ldots, 2\,T - 1$. Note that we do not add any information to $\boldsymbol{D}$ that is not included in the original data. Only the number of observations doubles, a fact that can easily be accounted for if one wants to estimate, e.g., the variance of the model outcome.

3 Dynamic Bayesian Network Classifiers

Before describing in detail how to build dynamic Bayesian classifiers, we briefly consider previous work. Friedman [9] pioneered with his Bayesian network approach to modelling gene interactions. A separate variable indicates the cell cycle phase, i.e., time. Husmeier [3] used a dynamic Bayesian network to model the lagged relations between genes using the likelihood of the graph as a scoring metric. Husmeier acknowledges the problem imposed by the limited size of available microarray time series. As earlier stated, we choose to predict the change in expression of individual genes by a classifier.

A probabilistic network classifier $M = (G, \boldsymbol{\theta})$ consists of a structural model specification, the directed graph G, and the parameters, $\boldsymbol{\theta}$, with the (un)conditional probability $\theta_{i,j,\pi(i)} = P(D_i = d_j \mid \pi(D_i) = \boldsymbol{d}_{\pi(D_i)})$. The notation $\pi(D_i) = \boldsymbol{d}_{\pi(D_i)}$ indicates the values of the parents of node D_i in the graph G (the parents constitute the nodes with arcs pointing directly to node D_i). Computation of the posterior probability distribution $P(C|\boldsymbol{X})$ is specified by the directed graph. It follows from the chain rule that the joint probability $P(\boldsymbol{d}) = P(c, \boldsymbol{x})$ is computed from

$$P(\boldsymbol{d}) = \prod_{i=1}^{k+1} P(D_i = d_j \mid \pi(D_i) = \boldsymbol{d}_{\pi(D_i)}) \tag{5}$$

A little manipulation of Bayes formula yields the posterior probability associated with class label c_j

$$P(c_j|\boldsymbol{x}) = \frac{P(c_j, \boldsymbol{x})}{\sum_m P(c_m, \boldsymbol{x})} \tag{6}$$

3.1 Learning Probabilistic Network Classifiers

Probabilistic network classifiers [10] have to be learned from a dataset $\boldsymbol{D}$. In the past, complete graphical models have successfully been learned using the approach introduced by Madigan & York [11]. However, their version of the MCMC-algorithm is not appropriate for learning probabilistic network classifiers, because it samples complete graphs drawn from the conditional distribution $P(G \mid \boldsymbol{D})$. Instead, we introduce a novel Markov Chain Monte Carlo technique based on the principles of Reversible Jump MCMC [12] to sample the posterior distribution probabilistic network classifiers. We make a simplification that leads to a less complex across-model sampling scheme than RJMCMC. Consequently, we can omit the Jacobian determinant term.

Let the variables in the learning database $\boldsymbol{D}$ be separated into a set of predictor variables X and a classification variable C, $D = (C, X)$. Our goal is to sample models from the following target distribution $P(L(C) \mid \boldsymbol{D})$, with $L(C)$ a score function (also called loss function [13])

$$P(L(C) \mid \boldsymbol{X}, G, \boldsymbol{\theta}^*, \boldsymbol{D}) = \prod_{\boldsymbol{d} \in \boldsymbol{D}} l(P(c \mid \boldsymbol{x}, G, \boldsymbol{\theta}^*, \boldsymbol{d}); v, \gamma) \tag{7}$$

with l the modified step function

$$l(y; v, \gamma) = \begin{cases} \gamma : & y \leq \frac{1}{2} - v \\ \frac{1}{2} : & |y - \frac{1}{2}| < v \\ 1 - \gamma : & y > \frac{1}{2} + v \end{cases} \tag{8}$$

for which it holds that $l(C = c) \in (0, 1)$, $\sum_c l(C = c) = 1$. The modified step function has two parameters, the *span of indeterminacy* v and the *bounding probability* γ. The parameter v determines the range of posterior probabilities regarded as ties, resulting in an intermediate score. The bounding probability $0.5 > \gamma > 0$ determines the gain or loss obtained by classifying a case correctly or wrongly, respectively. The score $L(C)$ can be considered a genuine probability, similar to the likelihood $P(\boldsymbol{D} \mid G)$ applied by Madigan & York. The distribution $P(L(C) \mid \boldsymbol{D})$ cannot be sampled directly, hence we perform the following factorization yielding a hierarchical Bayesian model:

$$\begin{aligned} &P(L(C) \mid \boldsymbol{D}) = \\ &\textstyle\sum_G P(L(C) \mid \boldsymbol{X}, G, \boldsymbol{D}) \\ &P(G \mid \boldsymbol{X}) P(\boldsymbol{X} \mid k, \boldsymbol{D}) P(k \mid \boldsymbol{D}) \end{aligned} \tag{9}$$

with G the directed graph, $\boldsymbol{X}$ the observations corresponding to the subset of selected predictor variables, and k the number of selected predictor variables. Computation of $P(L(C) \mid \boldsymbol{X}, G, \boldsymbol{D})$ requires a closed form solution to

$$P(L(C) \mid \boldsymbol{X}, G, \boldsymbol{D}) = \int_{\boldsymbol{\theta}} P(L(C) \mid \boldsymbol{X}, G, \boldsymbol{\theta}, \boldsymbol{D})\, P(\boldsymbol{\theta} \mid \boldsymbol{X}, G, \boldsymbol{D})\, d\boldsymbol{\theta} \tag{10}$$

in which $P(L(C) \mid \boldsymbol{X}, G, \boldsymbol{\theta}, \boldsymbol{D})$ is the probability of the score $L(C)$, given the parameter vector $\boldsymbol{\theta}$, the data associated with the predictor variables $\boldsymbol{X}$, the acyclic graph G and the database $\boldsymbol{D}$. As no closed form is presently available, we suggest to use instead $P(L(C) \mid \boldsymbol{X}, G, \boldsymbol{\theta}^*, \boldsymbol{D})$ with $\boldsymbol{\theta}^*$ the maximum-likelihood estimate of the parameter vector[1]. Note that the model G does not change as a function of $\boldsymbol{\theta}$. Since G does not depend on $\boldsymbol{\theta}$, conditioning on $\boldsymbol{\theta}^*$, the most likely parameter vector, will not strongly bias the estimate of $P(L(C) \mid \boldsymbol{X}, G, \boldsymbol{D})$. However, this approximation necessitates the use of a regularization prior. The following derivation is based on work presented elsewhere [14]. The variance of $\ln(L(C))$ equals the sum of the variances of $\ln(l(\boldsymbol{x}_i); v, \gamma)$, pertaining to the individual cases i

$$\sigma^2_{\ln(\,P(L(C))\,)} = \left(\ln\left(\frac{1}{2} + \gamma\right) - \ln\left(\frac{1}{2} - \gamma\right)\right)^2 \sum_i (p_i - p_i^2) \tag{11}$$

The probability p_i is in fact the probability per case that resampling the training set leads to the same winner resulting in $l(\boldsymbol{x}_i; v, \gamma) = 0.5 + \gamma$. Conversely, $1 - p_i$ is an error rate for a correctly classified case i. Consequently, we subtract $\sigma^2_{\ln(L(C))}$ from $\ln\{P(L(C) \mid \boldsymbol{X}, G, \boldsymbol{\theta}^*, \boldsymbol{D})\}$.

[1] This motivates our choice of score function in the first place.

The Markov Chain Monte Carlo algorithm should preferably not be biased towards a certain number of features or model complexity. Hence, we propose to use a uniform prior $P(k \mid \boldsymbol{D})$ on the size of the feature set k. For each feature set size k, each feature subset should be equally likely, so $P(\boldsymbol{X} \mid k, \boldsymbol{D})$ is also uniform. Finally, for a particular feature set, each possible model utilizing this feature subset should have the same prior, so $P(G \mid \boldsymbol{X})$ is uniform. We define the one-step look ahead neighbourhood of the graph G consisting of the directed acyclic graphs of classifiers that can be constructed by adding one arc to G or deleting one arc from G. The neighborhood $NB_C(G)$ is subdivided into four disjoint subsets

$$NB_C(G) = \{NB_C(G + 1_F), NB_C(G - 1_F), NB_C(G + 1_M), NB_C(G - 1_M)\} \quad (12)$$

The subset $NB_C(G + 1_F)$ contains the graphical models in $NB_C(G)$ where the addition of an arc implies that $G\prime$ contains one feature variable more than G. The subset $NB_C(G - 1_F)$ contains the models in $NB_C(G)$ where the deletion of an arc implies that $G\prime$ contains one feature variable less than G. The subset $NB_C(G + 1_M)$ contains the models in $NB_C(G)$ where the addition of an arc increases the complexity of $G\prime$, but where G and $G\prime$ include the same feature variables. $NB_C(G - 1_M)$ contains the models in $NB_C(G)$ where the deletion of an arc decreases the complexity of $G\prime$, but where G and $G\prime$ include the same feature variables. Define the appropriate proposal distribution q_C:

$$q_C(G \rightarrow G\prime) = \begin{cases} u < \frac{1}{4} & q_1(|NB_C(G + 1_F)|^{-1}) \\ \frac{1}{4} \leq u < \frac{1}{2} & q_2(|NB_C(G - 1_F)|^{-1}) \\ \frac{1}{2} \leq u < \frac{3}{4} & q_3(|NB_C(G + 1_M)|^{-1}) \\ \frac{3}{4} \leq u & q_4(|NB_C(G - 1_M)|^{-1}) \end{cases} \quad (13)$$

with $u \sim \mathcal{U}(0, 1)$. The proposals q_1, q_2, q_3 and q_4 result in a classifier pertaining to each of the four disjoint sub-neighborhoods, $NB_C(G + 1_F)$, $NB_C(G - 1_F)$, $NB_C(G + 1_M)$ or $NB_C(G - 1_M)$, respectively. The proposal distribution q_C implements the uniform priors, $P(G \mid \boldsymbol{X})$, $P(\boldsymbol{X} \mid k, \boldsymbol{D})$ and $P(k \mid \boldsymbol{D})$. So in each proposal, the MCMC-algorithm with the same probability chooses to add a feature, delete a feature, increase the model complexity or simplify the model (the two latter moves keep the same feature subset). The resulting Metropolis-Hastings ratio becomes

$$\frac{P(L(C) \mid \boldsymbol{X}_q, G_q, \boldsymbol{\theta}^*_q, \boldsymbol{D})\, P_q(\,(\boldsymbol{X}_q, k_q) \rightarrow (\boldsymbol{X}, k)\,)\, V}{P(L(C) \mid \boldsymbol{X}, G, \boldsymbol{\theta}^*, \boldsymbol{D})\, P_q(\,(\boldsymbol{X}, k) \rightarrow (\boldsymbol{X}_q, k_q)\,)\, V_q} \quad (14)$$

with q indicating the new proposal, the regularization terms

$$\ln(V_q) = -\alpha\, \sigma^2_{\ln(P(L(C)|X_q \cdots))} \quad \text{and} \quad \ln(V) = -\alpha\, \sigma^2_{\ln(P(L(C)|X \cdots))}.$$

The proposal probabilities, $P_q(\,(\boldsymbol{X}_q, k_q) \rightarrow (\boldsymbol{X}, k)\,)$ and $P_q(\boldsymbol{X}, k \rightarrow \boldsymbol{X}_q, k_q)$ correct for parts of the model space where one or more of the sub-neighborhoods are empty.

4 Experiments

To validate the applicability of our method on a true biological system, we used the yeast cell-cycle expression dataset from Spellman et al. [2]. The yeast cell cycle is a highly regulated process, with a central role for a class of genes named cyclins. Cyclins are transiently expressed in different phases of the cell-cycle, and team up with a cyclin-dependent kinase (CDK). Together, the cyclins and the kinases regulate the expression and/or activity of transcription factors, which in turn regulate the expression of genes that are directly involved in the diverse processes that prepare a yeast cell for division. We used an experiment where cells were initially synchronized, and subsequently followed in time as they progressed through the cell cycle.

The cyclins CLB2 and CLN3 are functional partners of the essential CDK CDC28. Clb2p/Cdc28p posttranscriptionally regulates transcription factors Mcm1p/Fkh2p through Ndd1p. We followed the expression of CDC28, CLB2, MCM1, and several target genes of MCM1/FKH2 to determine whether this genetic network could be identified using our method. Cln3p/Cdc28p are known to regulate the activity of the Swi5p transcription factor; their expression and the expression of target genes of Swi5p were analyzed. Ste12p, another transcription factor acting in concert with Mcm1p, was also analyzed in concert with some of its target genes.

To investigate the influence of our signal processing steps, parameter settings of the max-min filter and the scale-space transformation were varied, and co- and controlled regulatory events were compared to actual regulatory interactions described in the literature [15]. Finally, our co-regulatory relations were compared with the results obtained from hierarchical clustering (Euclidean distance measure). A summary of our results per target gene is presented in Table 1.

We varied the settings of the max-min filter and the scale-space transformation. In total 29 time points were sampled from the Spellman data. To obtain a data set with a uniformly sampled time, nearest neighbor linear interpolation was applied to a few time steps. This interpolation scheme was chosen because it can never introduce new extrema in the time series. Subsequently, dynamic predictor variables were extracted with lags ranging from 2, 1 and 0 time steps (with each time step corresponding to 10 minutes). As a complete time series with all three lags is required, only 27 time points were available. After preprocessing (Fig. 1), in total 54 (doubled) data points were available. The following genes were considered as targets: CLB1, BUD4, SWI4, CDC6, AGA1, ASH1, CDC45, CDC47, CTS1, FUS1 and MFA2. As predictive feature variables, the following variables were included: MCM1, STE12, CDC28, CLB2, CLN3 and SWI5. Corresponding to each target gene, the MCMC-algorithm was run 10.000 iterations. The most likely and second-most likely feature subsets occurring in the Markov chain were identified, see Table 1. We could find some co-regulations and controlled regulations with every setting applied. The max-min filter was important for the end result; when not applied, many spurious correlations were found, likely due to the relatively high noise in the signal corresponding to the lower expressed regulatory genes. The higher σ^2 was set, the more significant our results were. With σ^2 set at 4, only one false positive interaction $\mathcal{T}(\text{CLN3} \rightarrow \text{MFA2})$ was detected, yet some co-regulatory events were missed, that were apparent when σ^2 was set to 2. Since we were primarily interested in controlled regulation, we used the max-min filter set at

Table 1. Selected target genes listed in the most left column at different lag times are indicated by their names. Results consistent with co-regulation (also close in hierarchical clustering) or controlled regulation are indicated with a Y(es) in the 'valid' and 'close' columns, respectively. Spurious correlation is indicated with a N(o). The question marks indicate possible controlled regulations, where regulatory genes were co-regulated with their targets. The parentheses $(\ldots)$ indicate observations pertaining to the second-most likely model found by MCMC.

Target gene	Lag(0)	Valid	Close	Lag(-1, -2)	Valid
CLB1	CLB2	Y	Y		
BUD4	CLB2	Y	Y		
SWI4				CLB2	Y
CDC6				CLB2	Y
AGA1				CLB2	Y
ASH1				CLB2 (SWI5)	Y (Y)
CDC45				CLB2 (MCM1)	Y (Y)
CDC47	CDC28	N	N		
CTS1				SWI5	Y
FUS1	SWI5 (MCM1)	N (?)	N		
MFA2				CLN3 (CLB2)	N (Y)

$w = 3$, and chose σ^2 to be set at 2 in the scale space transformation. The regularization parameter α was set to 10.

5 Discussion

Our method relies solely on the timing of expression ratios of mRNAs, corresponding to the genes under investigation. It is possible to imagine that regulators, when altered in level, can change the level of their target genes at a given time in the near future. We expect the time course of regulatory events to be limited by diffusion of the molecules within the cell, and the rate of transcription of a target gene, and therefore we expect controlled regulations to occur within the time frame of minutes. Since one time point represents 7 minutes in the dataset under investigation, only the lag (0) co-regulation and lag (-1), and lag (-2) controlled regulatory events were taken into account.

Microarray data are inherently noisy, and only describe the expression of mRNA levels, ruling out the possibility to directly detect interactions due to cellular processes occurring after translation (e.g., mRNA decay, protein modifications, protein degradation). Despite these obstacles, our combination of preprocessing, coupled to selection of predictive features from a group of potential regulatory genes, allowed for robust detection of interactions.

The parameter settings of the max-min filter and the scale-space transformation had considerable influence on the genes detected in our method. Several factors can account

for this. Firstly, transcription factors are expressed at a low level, resulting in a higher variation in expression due to the inherent noise of microarray experiments. The max-min filter and the scale-space transformation both smoothen these smaller variations, resulting in a trade-off between noise suppression and sensitivity. Secondly, when a higher value for σ^2 is used in the scale-space transformation, a bias is introduced which can alter the timing of regulatory events. Finally, controlled regulations occurring within a time interval of seven minutes (the sampling time in the experiment) will be detected as co-regulations. Within the limits of the experimental set-up, we cannot catch these regulatory events, a shortcoming that could be circumvented by sampling at shorter time intervals.

We identified ten controlled regulatory events in the set of genes we analyzed, of which only one correlation turned out to be spurious. Two additional co-regulated genes ($\mathcal{C}(\mathrm{CDC28}, \mathrm{CDC47})$, $\mathcal{C}(\mathrm{MCM1}, \mathrm{FUS1})$) may represent controlled regulations characterized by shorter lags than the sampling time. It is interesting to note that the expression of these genes was distant in cluster analysis, a consequence of the expression ratios being inversed in sign, yet co-regulated. An example of the latter was $\mathcal{C}(\mathrm{MCM1}, \mathrm{FUS1})$, the high frequency and regular spacing of extrema lead us to conclude that the detected correlation was due to co-linearity, because a controlled stimulatory interaction is expected to show a lagged co-regulation with extrema being of the same sign. In the future we will include prior knowledge, such as whether a regulatory gene stimulates or represses transcription of a target gene, to circumvent this problem.

In summary, we present a proof of concept for a new method to extract regulatory interactions from microarray time series data. Despite the noisy character of the data and other experimental limitations, ten out of thirteen detected control-regulatory events corresponded to published experimental data, whereas one of the three false positives can be corrected using prior knowledge. Future approaches, incorporating knowledge about biological systems, will be expected to yield an even higher predictive accuracy.

6 Conclusion

In this article, we introduced a completely new approach to discovering putative regulative relations between genes studied in time series microarray experiments. The preprocessing steps make it possible to capture dynamic relations between sets of genes. Using Markov Chain Monte Carlo sampling and a new hierarchical Bayesian model, we discover control regulations and co-regulations between sets of genes. The method works well as it results in small compact graphs that reflect experimentally verified regulatory relations between genes. The predictive variables included in the (second) most likely graphs often exert control upon the target gene. Among 15 regulatory relations found, only 2 were spurious. In the future, we will evaluate our approach further on simulation data to get more insight into the parameter settings and on other real data sets to validate the method's appropriateness.

References

1. Datta, S., Datta, S.: Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics **19** (2003) 459–466

2. Spellman, P., Sherlock, G., M.Q., Z., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Molecular Biology of the Cell **9** (1998) 3273–3297
3. Husmeier, D.: Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. Bioinformatics **19** (2003) 2271–2282
4. Smith, V., Jarvis, E., Hartemink, A.: Evaluating functional network inference using simulations of complex biological systems. Bioinformatics **18** (2002) S216–S224
5. Verbeek, P., Vrooman, H., van Vliet, L.: Low-level image-processing by max min filters. Signal Processing **15** (1988) 249–258
6. Florack, L., ter Haar Romeny, B., Koenderink, J., Viergever, M.: Scale and the differential structure of images. Image and Vision Computing **10** (1992) 376–388
7. Lindeberg, T.: Scale-space for discrete signals. IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990) 234–254
8. Lindeberg, T., ter Haar Romeny, B. In: Linear scale-space II: Early visual operations. Kluwer Academic Publishers, Dordrecht (1994) 39–72
9. Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using bayesian networks to analyze expression data. Journal of Computational Biology **7** (2000) 601–620
10. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine learning **29** (1997) 131–163
11. Madigan, D., York, J.: Bayesian graphical models for discrete-data. International statistical review **63** (1995) 215–232
12. Green, P.: Reversible jump markov chain monte carlo computation and bayesian model determination. Biometrika **82** (1995) 711–732
13. Duda, R., Hart, P.: Pattern classification and scene analysis. John Wiley & Sons, New York (1973)
14. Egmont-Petersen, M., Feelders, A., Baesens, B.: Confidence intervals for probabilistic network classifiers. To appear in Computational Statistics and Data Analysis (2004)
15. Mendenhall, M., Hodge, A.: Regulation of cdc28 cyclin-dependent protein kinase activity during the cell cycle of the yeast *saccharomyces cerevisiae*. Microbiology and Molecular Biology Reviews **62** (1998) 1191–1243