Boulesteix, Strimmer:

Predicting Transcription Factor Activities from Combined Analysis of Microarray and ChIP Data: A Partial Least Squares Approach

Sonderforschungsbereich 386, Paper 411 (2005)

Online unter: http://epub.ub.uni-muenchen.de/

Predicting Transcription Factor Activities from Combined Analysis of Microarray and ChIP Data: A Partial Least Squares Approach

Anne-Laure Boulesteix and Korbinian Strimmer

Department of Statistics, University of Munich, Ludwigstr. 33, 80539 Munich, Germany.

February 4, 2005

Corresponding author:
Korbinian Strimmer
Phone: +49-89-21803225
Fax: +49-89-21805041
Email: korbinian.strimmer@lmu.de

Classification:
1. Physical Sciences, Statistics
2. Biological Sciences, Genetics

Manuscript information:

Number of text pages: 20	Number of words in Abstract: 180
Number of figures: 4	Total character count: 46970 (incl. figures and tables)
Number of tables: 2	

Abbreviations:
ChIP, chromatin immunoprecipitation; TFA, transcription factor activity;
PLS, partial least squares; NCA, network component analysis;

Abstract

The study of the network between transcription factors and their targets is important for understanding the complex regulatory mechanisms in a cell. However, due to post-translational modifications the regulator transcription levels (as measured, e.g., by microarray expression arrays) generally provide only little information about the true transcription factor activities (TFAs).

Here we propose an approach based on partial least squares (PLS) regression to infer true TFAs from expression data integrated with information from DNA-protein binding experiments (e.g., ChIP). This method is statistically sound also for a small number of samples and enables to detect functional interaction among the transcription factors themselves via the inference of “meta”-transcription factors. In addition, it allows to identify false positives in ChIP data as well as to predict activation and suppression activities (which is not possible from ChIP data alone). Subsequent to PLS inference, the estimated transcription factor activities may be subject to further analysis such as tests of periodicity or differential regulation. This method overcomes the limitations of previously used approaches, and is illustrated by analyzing expression and ChIP data from Yeast and *E. Coli* experiments.

Introduction

The transcription of genes is regulated by DNA binding proteins that attach to specific DNA promoter regions. These proteins are known as transcriptional regulators or transcription factors and recruit chromatin-modifying complexes and the transcription apparatus to initiate RNA synthesis[1;2].

In the last few years, considerable effort was produced by both experimental and computational biologists to identify transcription factors, their target genes and the sensitivity of the regulation mechanism to changes in environment[3;4;5]. An important technique for the identification of target genes bound in vivo by known transcription factors is the combination of a modified chromatin immunoprecipitation (ChIP) assay with microarray technology as proposed by Ren et al.[1]. For instance, in the budding yeast *Saccharomyces cerevisiae* ChIP experiments have been utilized to elucidate the binding interaction between 6270 genes and 113 preselected transcription factors[2]. However, as physical binding of transcription factors is only a necessary but not a sufficient condition for transcription initiation *ChIP data typically suffer from a large proportion of false positives*.

Several attempts have also been made to recover the network structure between transcription factors and their targets using only the gene expression levels of both the transcription factors and the targets, either with[6] or without[7] assuming a subset of putative regulators. Such approaches implicitly assume that the measured gene expression levels of the transcription factors reflect their actual activity. However, due to various complex post-translational modifications as well as due to interaction among transcription factors themselves, *regulator transcription levels are generally inappropriate proxies for transcription factor activities (TFA).*

In a few recent papers, integrative analysis of gene expression data and ChIP connectivity data has been suggested to overcome these issues[8]. Most prominently, Liao and coworkers have developed the technique of “network component analysis” (NCA)[9;10], a dimension reduction approach to *infer* the true regulatory activities. In NCA one can also incorporate further a priori qualitative knowledge about gene-transcription factor interactions[11]. Unfortunately, a major drawback of the

original NCA method is that for identifiability reasons it imposes very strong restrictions on the allowed network topologies which renders application of classic NCA difficult in many practical cases. Alter and Golub[12] introduced an approach to integrate ChIP and microarray data using pseudo-inverse projection. Like NCA, this method is based on an algebraic matrix decomposition (in this case singular value decomposition). However, this ignores measurement and biological error present in both connectivity and gene expression data. Kato et al.[13] proposed yet another integrative approach consisting of several steps combining sequence data, ChIP data and gene expression data. However, here gene expression is used only to check the coherence of expression profiles of genes with common sequence motifs, and not to estimate transcription factors activities. Finally, Gao et al.[14] suggested the "MA-Networker" algorithm which employs multivariate regression to estimate TFAs and backward variable selection to identify the active transcription factors. Unlike the other approaches, it fully takes account of stochastic error. However, for classical regression theory to be valid it is not only necessary that the number of gene targets is much greater than both the number of samples and the number of transcription factors, but also that the transcription factors are independent of each other. In particular the latter condition is clearly not generally satisfied with genome data.

Here, we suggest an alternative statistical framework to tackle the problem of network component and regulator analysis. Our approach centers around multivariate partial least squares (PLS) regression, a well-known analysis tool for high-dimensional data with many continuous response variables that has been widely applied, especially to chemometric data[15;16;17]. Using PLS we are not only able both to integrate and generalize previous NCA approaches, but also to overcome their respective limitations. In particular, PLS-based network component analysis offers a computationally highly efficient and statistically sound way to infer true TFAs for any given connectivity matrix. In addition, it allows to statistically assess the available connectivity information, and also to discover interactions and natural groupings among regulatory genes (corresponding to "meta"-transcription factors).

Methods

Network model.

Suppose gene expression data for n genes and m samples (= arrays, tissue types, time points etc.) are collected in a $n \times m$ data matrix $\tilde{\mathbf{Y}}$. Furthermore, let $\tilde{\mathbf{X}}$ denote the so-called connectivity matrix with n rows and p columns. Each column in $\tilde{\mathbf{X}}$ describes the strength of interaction between one of p transcription factors and the n considered gene targets. The entries of $\tilde{\mathbf{X}}$ can either be binary (0-1) or numeric (e.g. ChIP data), with a zero value indicating no physical binding between a transcription factor and a target.

In order to relate expression with connectivity data we consider the linear model

$$\tilde{\mathbf{Y}} = \mathbf{A} + \tilde{\mathbf{X}}\tilde{\mathbf{B}} + \mathbf{E}, \tag{1}$$

where $\mathbf{A}$ is $n \times m$ constant matrix, $\tilde{\mathbf{B}}$ is a $p \times m$ matrix of regression coefficients and $\mathbf{E}$ is a $n \times m$ matrix containing error terms. $\mathbf{A}$ contains the m different offsets, and $\tilde{\mathbf{B}}$ *may be interpreted as the matrix of the true transcription factor activities (TFAs) of the p transcription factors for each of the m samples.*

It it worthwhile to note that in this setting, unlike in most other gene expression analysis studies, the number of genes n is considered as the number of *cases* rather than the number of variables. In the present case the latter corresponds to the number of transcription factors p (hence in general $p < n$).

NCA and MA-Networker algorithms.

The above model linking TFAs both with gene expression of the regulated genes and external connectivity information has been the subject of a series of recent studies.

In the classic network component analysis approach[9;10] the offset matrix $\mathbf{A}$ is set to zero and the remainder of Eq. 1 is interpreted as dimension reduction that projects the output layer $\tilde{\mathbf{Y}}$ with m samples onto a "hidden" layer of $p < m$ transcription factors.

In the original NCA algorithm the coefficients $\tilde{\mathbf{B}}$ are obtained via a novel matrix decomposition that respects the constraints provided by the connectivity matrix $\tilde{\mathbf{X}}$. Unfortunately, this also imposes rather strict identifiability conditions. As a consequence, classic NCA may only be employed with certain classes of "NCA compatible" $\tilde{\mathbf{X}}$[9].

In contrast, the "MA-Networker" algorithm by Gao et al.[14] employs standard multiple least-squares regression in conjunction with step-wise variable selection to estimate the true transcription factor activities $\tilde{\mathbf{B}}$. This requires that the number of target genes is much larger than both the number of transcription factors and the number of samples. More important, however, is that the step-wise model selection procedure employed is only poorly suited if the regulator genes are themselves interacting with each other. This is a major drawback as it is biologically well-known that transcription factors often work in conjunction with other regulators, and rarely act independently.

Partial least squares regression.

Here we propose to employ the method of partial least squares regression[15;17] to inferring true TFAs and the functional interaction of regulators.

PLS is a well-known analysis tool for high-dimensional data with many continuous response variables that has been widely applied, especially to chemometric data[16]. PLS is particularly suited to the case of non-independent predictors and for small-sample regression settings. It is computationally highly efficient, it does not necessitate variable selection, and it additionally infers meaningful structural components.

For these reasons PLS is now being adopted as a standard tool for multivariate microarray data analysis, particularly in classification problems[18;19;20;21]. We believe that PLS also provides an excellent framework for integrative network analysis, as PLS combines dimension reduction with regression and variable selection, the two key elements from both the NCA and the MA-Networker approaches.

In a nutshell, the PLS algorithm consists of the following consecutive steps:

1. First, the data matrices $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ are centered to column mean zero, resulting in matrices $\mathbf{X}$ and $\mathbf{Y}$, in order to estimate and to remove the offset $\mathbf{A}$. In addition, it is common practice in PLS analysis (and also recommended here) to scale the input matrices to unit variance.

2. Second, using linear dimension reduction $\mathbf{T} = \mathbf{XR}$ the p predictors in $\mathbf{X}$ are mapped onto $c \leq \text{rank}(\mathbf{X}) \leq \min(p, n)$ latent components in $\mathbf{T}$ (an $n \times c$ matrix). See the section "SIMPLS algorithm" below for the precise procedure employed in this paper. *The important key idea in PLS is that the weights* $\mathbf{R}$ *(a* $p \times c$ *matrix) are chosen with the response* $\mathbf{Y}$ *explicitly taken into account, so that the predictive performance is maximal even for small* c*.*

3. Next, assuming the model $\mathbf{Y} = \mathbf{TQ}' + \mathbf{E}$, $\mathbf{Y}$ is regressed by ordinary least squares against the latent components $\mathbf{T}$ (also known as X-scores) to obtain the loadings $\mathbf{Q}$ (a $m \times c$ matrix), i.e. $\mathbf{Q} = \mathbf{Y}'\mathbf{T}(\mathbf{T}'\mathbf{T})^{-1}$.

4. Subsequently, the PLS estimate of the coefficients $\mathbf{B}$ in $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$ is computed from estimates of the weight matrix $\mathbf{R}$ and the Y-loadings $\mathbf{Q}$ via $\mathbf{B} = \mathbf{RQ}'$.

5. Finally, the coefficients $\tilde{\mathbf{B}}$ for the original Eq. 1 are computed by rescaling $\mathbf{B}$.

Note that it is step 2 that greatly distinguishes PLS from related bilinear regression approaches, such as principal and independent components regression (PCR/ICR) and the pseudo-inverse-based method by Alter and Golub[12]. In the latter approaches the scores $\mathbf{T}$ are computed solely on the basis of the data matrix $\mathbf{X}$ without considering the response $\mathbf{Y}$[16].

Other quantities often considered in PLS include, e.g., the X-loadings $\mathbf{P}$ that are obtained by regressing $\mathbf{X}$ against $\mathbf{T}$, i.e. $\mathbf{X} = \mathbf{TP}' + \mathbf{F}$ and $\mathbf{P} = \mathbf{Y}'\mathbf{T}(\mathbf{T}'\mathbf{T})^{-1}$.

SIMPLS algorithm.

PLS aims to find latent variables $\mathbf{T}$ that simultaneously explain both the predictors $\mathbf{X}$ and the response $\mathbf{Y}$. The original ideas motivating the PLS decomposition were

entirely heuristic. As a result, a broad variety of different but in terms of predictive power equivalent PLS algorithms have emerged – for an overview see, e.g., Martens[17].

For the present application to infer true TFAs we suggest to use the SIMPLS ("Statistically Inspired Modification of PLS") algorithm which has the following appealing properties[22;23;24]:

- it produces orthogonal, i.e. empirically uncorrelated, latent components,
- it allows for a multivariate response, and
- it optimizes a simple statistical criterion.

A further added advantage of SIMPLS is that it is also computationally more efficient than most other PLS algorithms. Note that other PLS variants are known in the literature that have predictive power equal to SIMPLS. However, these either provide orthogonal loadings rather than orthogonal latent components $\mathbf{T}$ (Martens' PLS), or they do not elegantly extend from 1-dimensional to m-dimensional response $\mathbf{Y}$ in terms of their optimized objective function (NIPALS).

In SIMPLS the latent components $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_c$ of the columns in $\mathbf{T}$ are inferred by sequentially estimating the column vectors $\mathbf{r}_1, \ldots, \mathbf{r}_c$ of $\mathbf{R}$ according to the following criterion[24]:

1. $\mathbf{r}_1$ is the unit vector (with $|\mathbf{r}_1| = 1$) maximizing the length $|\mathbf{Y}'\mathbf{X}\mathbf{r}_1|$ of the $m \times 1$ covariance vector $\mathrm{cov}(\mathbf{Y}, \mathbf{t}_1)$.
2. For all $j = 2, \ldots, c$, $\mathbf{r}_j$ are the unit vectors (with $|\mathbf{r}_j| = 1$) maximizing the length $|\mathbf{Y}'\mathbf{X}\mathbf{r}_j|$ of the vector $\mathrm{cov}(\mathbf{Y}, \mathbf{t}_j)$ subject to the orthogonality constraint $\mathbf{t}_i'\mathbf{t}_j = \mathbf{r}_i'\mathbf{X}_i'\mathbf{X}_j\mathbf{r}_j = 0$ for all $i = 1, \ldots, j-1$.

In the actual implementation of SIMPLS[22] the weights $\mathbf{R}$ and the derived quantities $\mathbf{T}$ and $\mathbf{Q}$ are obtained by a Gram-Schmidt-type procedure that constructs the desired orthogonal basis.

In our analysis we use the SIMPLS implementation as provided in the R package "pls.pcr" by Ron Wehrens (University of Nijmegen). In contrast the our definition given above his program returns orthonormal X-scores $\mathbf{T}^*$ and non-unit weights $\mathbf{R}^*$. For conversion define $\mathbf{M} = \text{diag}(|\mathbf{r}_1^*|, \ldots, |\mathbf{r}_c^*|,)$ and set $\mathbf{T} = \mathbf{T}^*\mathbf{M}^{-1}$, $\mathbf{R} = \mathbf{R}^*\mathbf{M}^{-1}$, $\mathbf{Q} = \mathbf{Q}^*\mathbf{M}$, and $\mathbf{P} = \mathbf{P}^*\mathbf{M}$. This provides orthogonal scores and unit-norm weights.

The resulting estimates of the matrices $\mathbf{B}$, $\mathbf{T}$, and $\mathbf{R}$ are now straightforward to interpret in terms of transcriptional regulation. $\mathbf{B}$ (and $\tilde{\mathbf{B}}$) give the inferred activities of the p transcription factors in each of the m experiments. The inferred latent components $\mathbf{T}$ describe "meta"-transcription factors that combine related groups of transcription factors. $\mathbf{R}$ reflects the involvement of each of the p regulators in the c meta-factors.

Determining the number of PLS components.

A remaining aspect of PLS regression analysis is the optimal choice of the number c of latent components. If the maximal value $c_{\text{max}} = \text{rank}(\mathbf{X})$ is chosen, then PLS becomes equivalent to principal components regression (PCR) with the same number of components, and if additionally $n > p$ both PLS and PCR turn into ordinary least-squares multiple regression.

Hence, with PLS it is desirable to choose as small a value of c as possible without sacrificing too much predictive power. One straightforward statistical procedure to estimate this minimum value c_{min} is the method of cross-validation, which proceeds as follows:

1. Split the set of n genes randomly into 2 sets: a learning set containing $2/3$ of the genes and a test set containing the remaining genes.
2. Use the learning set to determine the matrix of regression coefficients $\mathbf{B}$ for different values $c = 1, 2, \ldots, c_{\text{max}}$.
3. Predict the gene expression of the $n/3$ genes from the test set using $\mathbf{B}$ with the different values of c.

4. Repeat steps 1-3 $K = 100$ times and compute the mean squared prediction error for each c.

Subsequently, the value of c yielding the smallest mean squared prediction error is selected. The results of this procedure can also be visualized graphically (see Figure 1 below for an example with real data).

Alternatively, the optimal number of components may also be determined by considering the value of the criterion $Z_i = |\mathbf{Y}'\mathbf{t}_i|$ for a given latent component $\mathbf{t}_i$. If Z_i falls below an a priori specified threshold then $c_{\text{min}} = i$ is reached.

Results

Data sets

Next, we illustrate the versatility of the proposed PLS approach to network component analysis by analyzing several real biomolecular data sets.

First, in order to validate the linear regression approach (Eq. 1) we reanalyzed hemoglobin data by Liao et al.[9] . Second, we analyzed two different *S. Cerevisae* gene expression data sets in conjunction with a regulator-target connectivity matrix from the large-scale ChIP experiment of Lee et al.[2]. The investigated yeast expression data comprise a time series experiment[25] and a compilation of yeast stress response experiments[26;6]. Finally, we analyzed expression and connectivity data for an *E. Coli* regulatory network containing 100 genes and 16 transcription factors[10]. The general characteristics of these four data sets are summarized in Table 1.

TABLE 1 ABOUT HERE

The investigated data were preprocessed as follows. The yeast ChIP data set[2] contains protein-DNA interaction data for 6270 genes and 113 transcription factors. It includes missing values that correspond to non-interacting gene-transcription factor pairs. Although ChIP data are essentially continuous, it is common practice to dichotomize the data according to the p-values into discrete levels of interaction (0 or 1). In this study, we used the data obtained at a p-value threshold of 0.001, as suggested by Lee et al.[2]. However, note that in contrast to the NCA method, the dichotomization of the ChIP data is optional in our approach.

The Spellman et al.[25] microarray data originally contains the gene expression of 4289 genes at 24 time points during the cell-cycle. From these genes, a subset of 3638 are also contained in the Lee et al.[2] ChIP data set. Our analysis is based on these 3638 genes. Similarly, the Segal-Gasch expression data set[26;6] contains the gene expression of 2292 genes for 173 arrays corresponding to different stress conditions (e.g., heat shock, amino acid starvation, nitrogen depletion). From 2292 genes a subset of 1993

overlap with the genes considered in the ChIP data.

The connectivity matrix for the *E. coli* data was compiled mainly by Kao et al.[10] from the RegulonDB[11] database, in addition they also incorporated a few corrections using literature data. The temporal *E. coli* expression data for 100 genes across 25 time points was introduced in Kao et al.[10] and is publicly available at `http://www.seas.ucla.edu/~liaoj/`.

Validation of the regression approach

The hemoglobin data used in Liao et al.[9] for validation of the classic NCA approach have the advantage that the true coefficients $\tilde{\mathbf{B}}$ of the network model in Eq. 1 are known, and therefore can be directly compared with the inferred values.

Reanalyzing this data we showed that the true regression coefficients can be recovered exactly by multivariate regression (of which PLS is a special case). According to Liao et al.[9] this is also true for classic NCA but not for PCA and ICA interpretations of Eq. 1. This can be explained by the fact the both PCA and ICA do not explicitly take account of the response $\mathbf{Y}$, whereas NCA and PLS do.

PLS components and Y-loadings

Subsequently, we determined the minimum number of PLS components for the yeast and *E. coli* data sets using cross-validation. The results are plotted in Figure 1 (top) after normalization (the mean cross-validation error with one PLS component is set to 1). As can be seen from Figure 1, the minimal mean cross-validation error is obtained with 5 PLS components for the Spellman data, 8 PLS components for the Gasch-Segal data and 2 PLS components for the *E. coli* data. For comparison, the (normalized) objective criterion $|\mathbf{Y}'\mathbf{t}_i|$ of the SIMPLS algorithm is also represented on Figure 1 (bottom) for different numbers of PLS components. These results are in good agreement with the cross-validation error: the cross-validation error increases when PLS components with a low objective criterion are added.

FIGURE 1 ABOUT HERE

The Y-loadings contained in the $m \times c$ matrix $\mathbf{Q}$ give the projection of the c "meta"-transcription factors for each of the m experiments. As can be seen from Figure 2 for the Spellman data, both the first and the third meta-factors explain the periodic part of the expression data, but with different phases. The second meta-factor corresponds to small oscillations with very short period, whereas the fourth and the fifth meta-factors reflect long-time trends (slow and step-wise increasing, respectively). Using Fisher's g-test as proposed in Wichert et al.[27], we detected statistically relevant periodicity for the four first meta-factors. In Figure 2, the Y-loadings are also represented for the *E. coli* data. Whereas the projection of the first meta-factor is approximately constant over time, the projection of the second meta-factor increases strongly and (almost) uniformly. Thus, in both data sets, the PLS algorithm allows to extract from the data meta-factors corresponding to distinct latent trends.

FIGURE 2 ABOUT HERE

For the Gasch-Segal data, the m experiments do not correspond to different time points but to 13 different stress conditions (see Gasch et al.[26] for further details, and Table 2 for the list of the conditions). In this case the Y-loadings may be interestingly analyzed using Wilcoxon's rank sum test. For each condition k and each meta-factor j, we tested the H_0 hypothesis that the median of the projection of the j-th meta-factor is the same in condition k as in all the other conditions ($\{1, \ldots, k-1, k+1, \ldots, 13\}$). In this situation, Wilcoxon's rank sum test is preferable to the well-known two-sample t-test, because some of the conditions include only a very small number of experiments. The results obtained with a p-value threshold of 0.05 are displayed in Table 2. The entries 1 and 0 correspond to significant and unsignificant (FDR adjusted) p-values, respectively. As can be seen from Table 2, each PLS component carries a particular pattern of associated significant conditions, indicating that the meta-factors capture a distinct *direction* of the data.

TABLE 2 HERE

Inferred transcription factor activities

One of the main objectives of our PLS-based approach is to estimate the true transcription factor activities (TFAs). Although all the TFAs can be estimated in the same way for the three data sets, we display only the evolution over time of a few interesting TFAs for the two time series data sets Spellman and *E. coli*.

The TFAs (top) and expression profiles (bottom) of 4 well-known cell-cycle regulators are depicted in Figure 3 for the Spellman data. The TFAs of MCM1, SWI4, SWI5 and ACE2 show highly periodic patterns, which is consistent with common biological knowledge. In contrast, the *expression* profiles of MCM1 and SWI4 are not periodic (this can be confirmed by Fisher's g-test[27]). On the other hand, the expression profiles of SWI5 and ACE2 are periodic, however not with the same phase as the inferred TFAs. This may indicate either inhibiting or a phase-shift effect of the transcription factors on the regulated genes.

FIGURE 3 ABOUT HERE

The remainder of the TFAs and the regulated genes were also tested for periodicity with the g-test[27]. After FDR adjustment of the p-values, we obtained that 62 of the 113 transcription factors (= 55%) in the Spellman/Lee data have significantly periodic TFAs at the level 0.05. In contrast, only 804 of the 4289 genes (= 19%) exhibit significantly periodic expression profiles.

For the *E. coli* data the time profiles of the estimated TFAs of the 16 transcription factors are represented in Figure 4. The TFAs of ArcA, GatR, Lrp, PhoB, PurR, RpoS decrease over time, the TFAs of CRP, CysB, FadR, IcIR, NarL, RpoE, TrpR and TyrR remain approximately constant and the TFAs of FruR and LeuO increase strongly. This is consistent with previous results obtained by NCA[10]. We point out, however, that unlike NCA our approach may be applied to any arbitrary network topology,

whereas the present *E. coli* network was chosen specifically to meet the NCA compatibility criteria[9].

FIGURE 4 ABOUT HERE

As can be seen already from the few examples depicted in Figure 3, the TFAs do not always correlate with the respective expression profiles. We tested this for all the transcription factors whose expression profile was also included in the data sets. For the Segal-Gasch data, we found that only 63 from the 90 available transcription factors exhibit expression profiles that are correlated with TFAs (at the level 0.05 with FDR p-value adjustment). For the Spellman time series data none of the 78 available TFA-expression profile pairs are correlated. These results clearly indicate that methods investigating transcriptional regulation with expression data as their sole basis are likely to miss potentially important regulation activities.

Gene-regulator coupling factors

Another topic of interest is the identification of false positives in ChIP data. Following Gao et al.[14] we investigate this problem via Pearson's correlation test. For each supposed gene-transcription factor pair (according to the dichotomized ChIP data) we test if the inferred TFA is significantly correlated with the expression profile of the regulated gene. For the Segal-Gasch data, we obtain that 73% of the 1495 gene-transcription factor pairs are correct (i.e. the TFA is significantly correlated with the expression profile at the level 0.05 with FDR p-value adjustment). The concordance with the ChIP connectivity information is much worse for the Spellman data where only 32% of the 2535 gene-transcription factor pairs are significantly correlated. Note that the false positive rates as obtained above actually constitutes an underestimation, since the TFAs are estimated regression coefficients, and even if all the pairs were false positives, some of them would still yield a high correlation.

Discussion

Network component analysis combines microarray data with ChIP data with the aim to enhance the estimation of regulator activities and of connectivity strengths. In this paper we have presented an approach to NCA based on partial least squares, a computationally efficient statistical regression tool.

Our PLS framework allows to overcome several drawbacks inherent both in the classic NCA methods based on matrix decomposition and in the MA-Networker algorithm. Its simplicity (no iterative step, no variable selection, no stochastic search) and its flexibility (no distributional assumptions, no topological constraints, no conditions on the dimensions) compared to competing approaches make it particularly attractive as an integrative method for analyzing complex regulatory networks. Moreover, the PLS algorithm not only extracts information on gene-regulator and on TFA-expression profile pairs but also identifies coherent meta-factors reflecting the main directions of variation of the data, taking account both of the expression ($\tilde{\mathbf{Y}}$) and the connectivity information ($\tilde{\mathbf{X}}$).

Our analysis of biological data shows the versatility of our PLS approach and at the same time dramatically confirms the necessity of a combined expression-ChIP analysis for regulatory inference. Particularly striking are the in part drastic differences between the measured transcription levels, and the PLS-inferred transcription activities. According to Segal et al.[6] some transcription factors may also not be active in all conditions. Note that this assumption is also automatically taken into account by our approach.

NCA in general, and the present PLS-based variant in particular, may be criticized for relying on a simple linear model. While biological analysis to a large extent validates that assumption, more elaborate regression approaches such as generalized linear models (GLMs) or generalized additive models (GAMs) are conceivable that, combined with PLS, may potentially even further enhance our current understanding of the complex structures governing genetic networks.

Acknowledgments

We thank Eran Segal for providing the complete *Saccharomyces cerevisiae* data set[6] and James Liao for providing the hemoglobin data[9]. This work was supported by the Deutsche Forschungsgemeinschaft (DFG) through an Emmy-Noether research grant to K.S. and the Sonderforschungsbereich 386.

Appendix: Computer program

A computer program (written in the R language[28]) for estimating true transcription factor activities with PLS is available from the authors' homepage.

References

[1] B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Volkert, C. J. Wilson, S. P. Bell, and R. A. Young. Genome-wide location and function of dna binding proteins. *Science*, 290:2306–2309, 2000.

[2] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298:799–804, 2002.

[3] V. R. Iyer, C. E. Horak, C. S. Scafe, D. Botstein, M. Snyder, and P. O. Brown. Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature*, 409:533–538, 2001.

[4] B. van Steensel, J. Delrow, and H. J. Bussemaker. Genomewide analysis of drosophila gaga factor target genes reveals context-dependent DNA-binding. *Proc. Natl. Acad. Sci. USA*, 100:2580–2585, 2003.

[5] C. T. Harbison, D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. Macisaac, T. W. Danford, N. M. Hannett, J. B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel, and R. A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431:99–104, 2004.

[6] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifing regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34: 166–176, 2003.

[7] M. Xiong, J. Li, and X. Fang. Identification of genetic networks. *Genetics*, 166: 1037–1052, 2004.

[8] Z. Li and C. Chan. Extracting novel information from gene expression data. *Trends Biotechnol.*, 22:381–383, 2004.

[9] J. C. Liao, R. Boscolo, Y. L. Yang, L. M. Tran, C. Sabatti, and V. P. Roychowdhury. Network component analysis: reconstruction of regulatory signals in biological systems. *Proc. Natl. Acad. Sci. USA*, 100:15522–15527, 2003.

[10] K. C. Kao, Y.-L Yang, R. Boscolo, C. Sabatti, V. Roychowdhury, and J. C. Liao. Transcriptome-based determination of multiple transcription regulator activities in Escherichia coli by using network component analysis. *Proc. Natl. Acad. Sci. USA*, 101:641–646, 2004.

[11] H. Salgado, A. Santos-Zavaleta, S. Gama-Castro, D. Millan-Zarate, E. Diaz-Peredo, F. Sanchez-Solano, E. Perez-Rueda, C. Bonavides-Martinez, and J. Collado-Vides. RegulonDB (version 3.2): transcriptional regulation and operon organization in escherichia coli K-12. *Nucleic Acids Res.*, 29:72–74, 2001.

[12] O. Alter and G. H. Golub. Integrative analysis of genome-scale data by using pseudoinverse projection predicts novel correlation between DNA replication and RNA transcription. *Proc. Natl. Acad. Sci. USA*, 101:16577–16582, 2004.

[13] M. Kato, N. Hata, N. Banerjee, B. Futcher, and M. Q. Zhang. Identifying combinatorial regulation of transcription factors and binding motifs. *Genome Biology*, 5:R56, 2004.

[14] F. Gao, B. C. Foat, and H. J. Bussemaker. Defining transcriptional networks through integrative modeling of mRNA expression and trasncription factor binding data. *BMC Bioinformatics*, 5:31, 2004.

[15] S. Wold, H. Martens, and H. Wold. The multivariate calibration method in chemistry solved by the PLS method. In A. Ruhe and B. Kagstrom, editors, *Proc. Conf. Matrix Pencils*, Lecture Notes in Mathematics, pages 286–293. Springer Verlag, Heidelberg, 1983.

[16] I. E. Frank and J. H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35:109–135, 1993.

[17] H. Martens. Reliable and relevant modelling of real world data: a personal account of the development of PLS regression. *Chemom. Intell. Lab. Syst.*, 58: 85–95, 2001.

[18] S. Datta. Exploring relationships in gene expressions: a partial least squares approach. *Gene Expression*, 9:249–255, 2001.

[19] D. Nguyen and D. M. Rocke. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18:39–50, 2002.

[20] D. Nguyen and D. M. Rocke. Partial least squares proportional hazard regression for application to DNA microarray survival data. *Bioinformatics*, 18: 1625–1632, 2002.

[21] A.-L. Boulesteix. PLS dimension reduction for classification with microarray data. *SAGMB*, 3:33, 2004.

[22] S. de Jong. SIMPLS: an alternative approach to partial least squares regression. *Chemom. Intell. Lab. Syst.*, 18:251–253, 1993.

[23] C. J. F. Ter Braak and S. de Jong. The objective function of partial least squares regression. *J. Chemometrics*, 12:41–54, 1998.

[24] S. de Jong, B. M. Wise, and N. L. Ricker. Canonica partial least squares and continuum power regression. *J. Chemometrics*, 15:85–100, 2001.

[25] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9:3273–3297, 1998.

[26] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, 11:4241–4257, 2000.

[27] S. Wichert, K. Fokianos, and K. Strimmer. Indentifying periodically expressed transcripts in microarray time series data. *Bioinformatics*, 20:5–20, 2004.

[28] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2004. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

Table 1. Characteristics of the analyzed data sets.

Data	Reference	n	p	m	$c_{\min}$
Hemoglobin	Liao et al.[9]	7	3	321	3
S. cerevisiae	Spellman et al.[25]	3638	113	24	5
S. cerevisiae	Segal et al.[6]; Gasch et al.[26]	1993	113	173	8
E. coli	Kao et al.[10]	100	16	23	2

Abbreviations: n, number of genes; p, number of transcription factors; m, number of arrays resp. measurements.

Table 2. Significant conditions for the first 8 PLS components of the Segal-Gasch yeast data set.

Condition \ PLS Component	1	2	3	4	5	6	7	8	Arrays
Heat shock	0	0	0	0	0	0	0	0	1-9,12-15
Variable temperature shocks	0	0	1	0	1	0	0	0	21-25
Hydrogen peroxide	0	0	0	0	0	1	0	0	36-45
Menadione	0	1	0	0	1	1	0	0	46-54
DTT	0	0	0	0	0	0	0	0	55-69
Diamide	1	1	1	0	0	0	1	1	70-77
Sorbitol osmotic shock	0	0	0	0	0	0	0	0	78-89
Amino acid starvation	0	0	1	1	1	0	1	1	91-95
Nitrogen depletion	0	0	1	0	0	1	1	1	96-105
Diauxic shift	0	0	1	0	0	0	1	0	106-112
Stationary phase	1	1	0	1	1	1	1	0	113-134
Continuous carbon sources	1	0	0	0	0	1	0	1	148-160
Continuous temperatures	1	0	0	0	0	0	1	0	161-173

Figure Legends

Figure 1. *Top row:* Mean sum of squared prediction error for *E. Coli* and yeast data sets over 100 cross-validation runs. *Bottom row:* maximized objective criterion for each PLS component.

Figure 2. Y-loadings for the *E. Coli* (*top row*) and Spellman (*bottom row*) data sets.

Figure 3. Time profiles of the TFAs (*top row*) of four well-known cell-cycle transcription factors from the Spellman data compared to the respective gene expression measurements (*bottom row*).

Figure 4. Time profiles of the 16 estimated TFAs (*E. Coli* data).

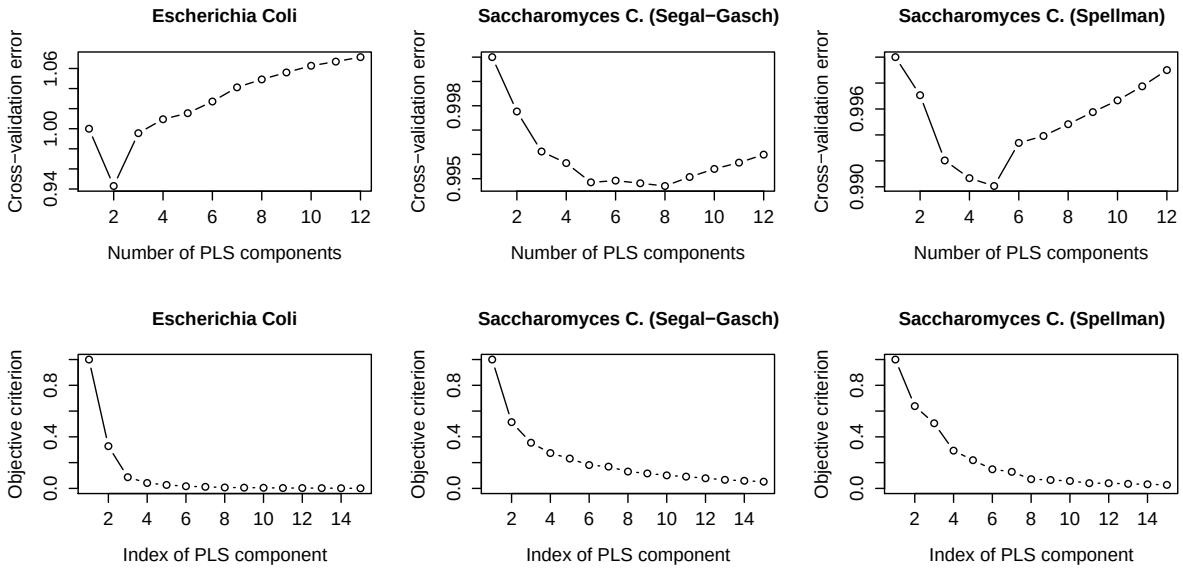
Escherichia Coli
Saccharomyces C. (Segal−Gasch)
Saccharomyces C. (Spellman)
Cross−validation error
0.94
1.00
1.06
0.995
0.998
0.990
0.996
2
4
6
8
10
12
Number of PLS components
Objective criterion
0.0
0.4
0.8
14
Index of PLS component

Figure 1:

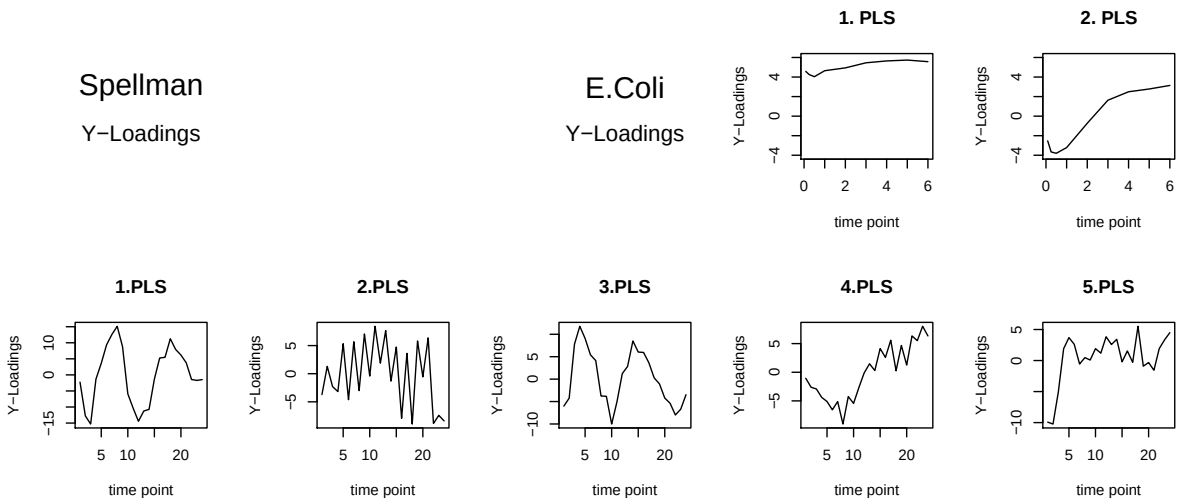
Spellman
Y−Loadings
E.Coli
Y−Loadings
1. PLS
2. PLS
Y−Loadings
time point
1.PLS
2.PLS
3.PLS
4.PLS
5.PLS

Figure 2:

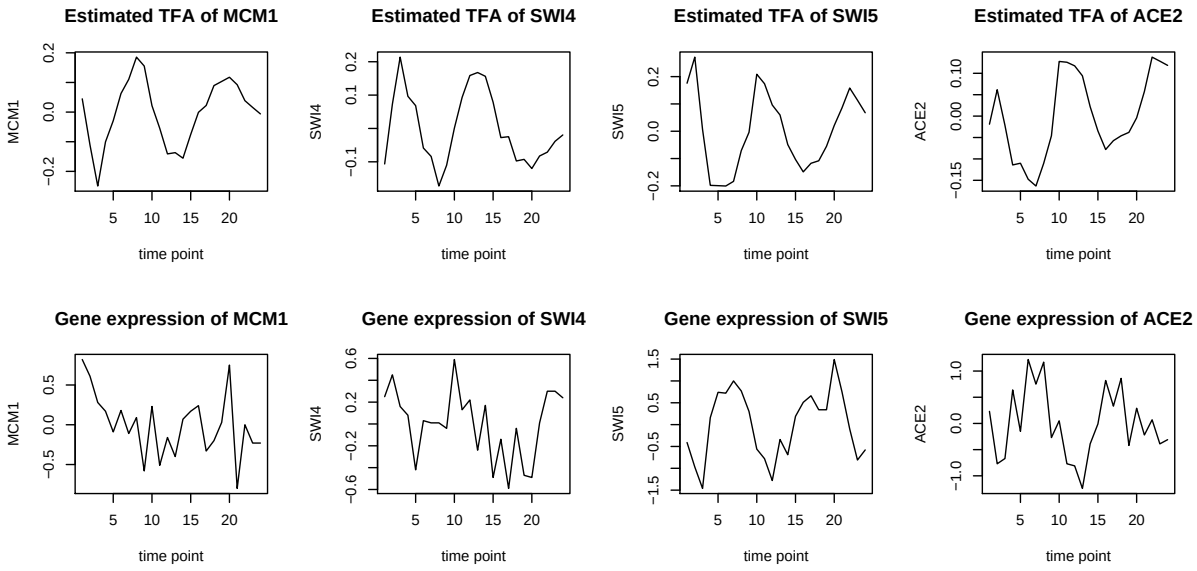
Estimated TFA of MCM1
Estimated TFA of SWI4
Estimated TFA of SWI5
Estimated TFA of ACE2
MCM1
SWI4
SWI5
ACE2
time point
Gene expression of MCM1
Gene expression of SWI4
Gene expression of SWI5
Gene expression of ACE2

Figure 3:

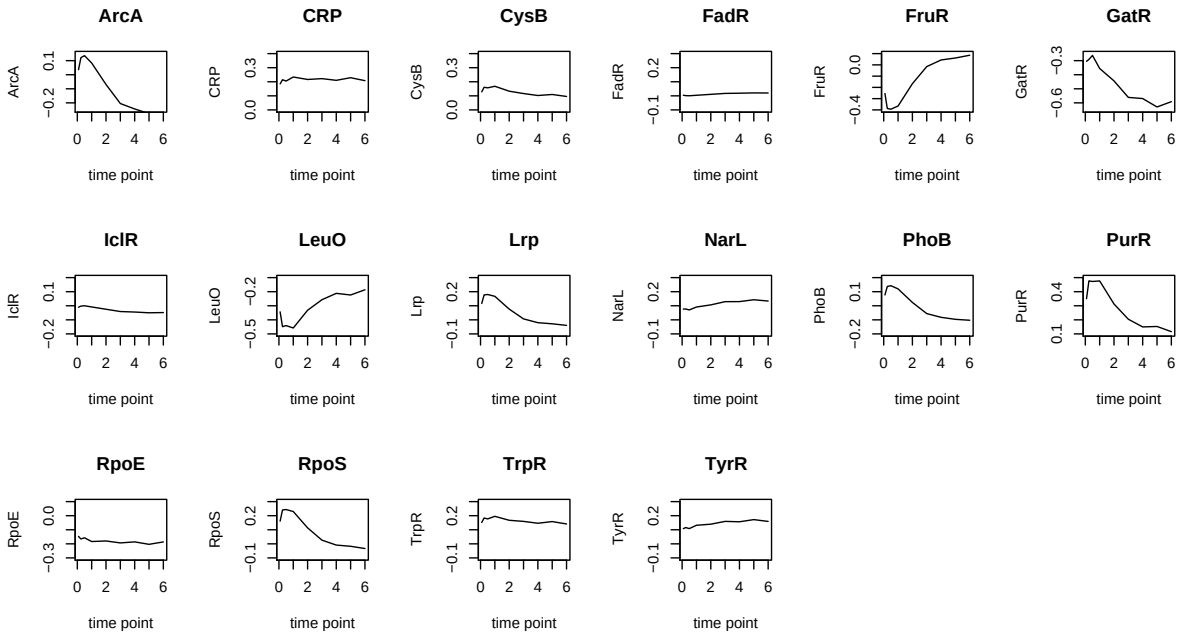
ArcA
CRP
CysB
FadR
FruR
GatR
IclR
LeuO
Lrp
NarL
PhoB
PurR
RpoE
RpoS
TrpR
TyrR
time point

Figure 4: