Cross-species common regulatory network inference without requirement for prior gene affiliation

Amin Moghaddas Gholami\textsuperscript{1,2} and Kurt Fellenberg\textsuperscript{1,*}

\textsuperscript{1}Chair of Proteomics and Bioanalytics, Center for Integrated Protein Sciences Munich (CIPSM), Technische Universität München, Emil Erlenmeyer Forum 5, 85354 Freising and \textsuperscript{2}Functional Genome Analysis, German Cancer Research Center (DKFZ), Im Neuenheimer Feld 580, 69120 Heidelberg, Germany

\textbf{ABSTRACT}

\textbf{Motivation:} Cross-species meta-analyses of microarray data usually require prior affiliation of genes based on orthology information that often relies on sequence similarity.

\textbf{Results:} We present an algorithm merging microarray datasets on the basis of co-expression alone, without any requirement for orthology information to affiliate genes. Combining existing methods such as co-inertia analysis, back-transformation, Hungarian matching and majority voting in an iterative non-greedy hill-climbing approach, it affiliates arrays and genes at the same time, maximizing the co-structure between the datasets. To introduce the method, we demonstrate its performance on two closely and two distantly related datasets of different experimental context and produced on different platforms. Each pair stems from two different species. The resulting cross-species dynamic Bayesian gene networks improve on the networks inferred from each dataset alone by yielding more significant network motifs, as well as more of the interactions already recorded in KEGG and other databases. Also, it is shown that our algorithm converges on the optimal number of nodes for network inference. Being readily extendable to more than two datasets, it provides the opportunity to infer extensive gene regulatory networks.

\textbf{Availability and Implementation:} Source code (MATLAB and R) freely available for download at http://www.mchips.org/supplements/moghaddasi_source.tgz

\textbf{Contact:} kurt@tum.de

\textbf{Supplementary information:} Supplementary data are available at Bioinformatics online.

Received on October 19, 2009; revised on February 24, 2010; accepted on February 25, 2010

1 INTRODUCTION

Microarray technique, albeit barely older than a decade, is now both mature and ubiquitous, accumulating an unprecedented amount of quantitative genome wide information (Quackenbush, 2006). Although each study is conducted to generate valuable insights in and of itself, it becomes more and more desirable to put them into a larger context. Various meta-analysis techniques combine individual studies conducted by different authors. Choi et al. (2003) and Rhodes et al. (2002) were among the first authors to introduce meta-analysis for microarray data. Most meta-analysis studies have been performed on cancer (Ma and Huang, 2009; Marot et al., 2009). In this field of research, biomaterial is often limited, in other cases the price for the microarrays is the bottleneck. In typical microarray-based studies, tens of thousands of genes (variables) are only investigated across (at most) hundreds of biological samples (observations). The asymmetry of the data tables poses a problem for inferring gene regulatory networks (GRN) by reverse engineering (Ramasamy et al., 2008). Alleviating the asymmetry by combining the datasets therefore largely increases their use for systems biology.

A multitude of algorithms have been reported for network inference from gene expression data. Dynamic Bayesian networks (DBN) capture conditional independence between variables. They are relatively easy to interpret (Friedman et al., 2000) yielding directed graphs. They can handle noisy data and even feedback loops in biological systems (Smith et al., 2002). In order to maximize the number of samples accounted for in one GRN reverse engineering step, it seems preferable for any method to first combine the data instead of combining the resulting networks later on. Several methods can be applied to this end (Conlon et al., 2007; Garrett-Mayer et al., 2008; Gilks et al., 2005; Rhodes et al., 2004; Stevens and Doerge, 2005; Wang et al., 2004; Yang and Sun, 2007). However, all of these methods take as input the affiliation of genes between the datasets.

When combining data stemming from different species, sequence homology can be used to affiliate orthologs. However, due to the ambiguity of orthology relations, mapping across species is challenging. Lineage-specific gene duplications can give rise to a different number of paralogs in one species compared to another species. One cannot tell which paralog (or in-paralog) retains the function of the ancestral gene or has been co-opted into a new function.

We present a way of combining datasets that does not need any genes (or samples) to be affiliated beforehand. While such information can be easily incorporated to assist the process, our algorithm also performs well without being provided with any affiliations, purely driven by coherences among the data. That opens the door to fully automated combining and modeling of all microarray datasets accumulated to date.

2 METHODS

2.1 Datasets and pre-processing

To introduce the method, we show its application to two yeast cell cycle studies (Rustici et al., 2004; Spellman et al., 1998) comprising
2.2 Co-inertia analysis

Co-inertia analysis (CIA) is a multivariate approach that can identify co-
relationships within multiple datasets by finding successive principal axes of
maximum co-variance. It was first introduced applying ecological data
(Dolédec and Chessel, 1994) using co-inertia as a measure of co-structure
between two data matrices. When the matrices are centered, co-inertia
is a sum of square covariances. A formal definition is provided in the
Supplementary data (S2.1).

Culhane and co-workers demonstrated the efficiency of CIA on cross-
platform comparisons of gene expression data (Culhane et al., 2003). CIA
is often used in combination with principal component analysis (PCA) or
correspondence analysis (CA), the latter being capable of visualizing genes
and hybridizations at the same time (Fellenberg et al., 2001). While, as
with PCA, similarity among genes as well as similarity among hybridizations
is depicted as proximity, a gene that is particularly up-regulated under
a certain condition will be located in the direction of this condition. The
farther away from the centroid in this direction (towards the outer margin
of the plot) it is displayed, the stronger the association (Culhane et al.,
2003; Fellenberg et al., 2001). If used together with CIA, genes and
hybridizations are shown simultaneously for both datasets, projecting their
common variance or co-inertia (Supplementary Fig S2). Here, proximity
among objects and directions can be interpreted as aforementioned, now
highlighting common trends and patterns. Overall similarity of the datasets
is captured by the RV-coefficient (RV) that is a commonly used matrix
distance measure (e.g., the RV-coefficient). While, as with PCA, all
matrices are centered, the RV-coefficient is computed as the sum of
squared differences between the objects in each data matrix.

Prerequisite for CIA is that either the genes or the hybridizations are
affiliated between the two datasets. Therefore, either the columns or the rows
of the tables must match (and have equal weights). In the following text,
we refer to the variables (genes or hybridizations) needed to be affiliated
beforehand as ‘connecting variables’ and to the distances between objects
in a CIA output (projection) as ‘projected distances’. We used Hungarian
algorithm to affiliate connecting variables in CIA.

2.3 Matching by Hungarian algorithm

Two sets of objects (here genes or hybridizations from the two datasets to be
combined) can be matched by the Hungarian algorithm, also called Kuhn–
Munkres algorithm (Kuhn, 1955). It takes as input a penalty weight matrix
of all possible pairwise projected distances and computes the pairs summing
up to minimal penalty (Supplementary data S3 and Fig S1). The original
publication refers to a quadratic penalty matrix. However, the Hungarian
algorithm can also be applied to sets of different cardinalities by adding
virtual objects of highest penalty to the smaller set until its cardinality
matches the larger one (Bourgeois and Lassalle, 1971). Here, virtual genes
or samples) have been added to the penalty matrix showing the maximum
of all occurring pairwise projected distances to all other genes (samples).

2.4 k-means clustering

Data can be subdivided into pre-defined numbers of homogeneous gene or
sample (array) clusters by the k-means algorithm. Here, we performed it on
the $x$-distance, the same distance measure that governs CIA.

2.5 Back-transformation

CIA projection reduces the dimensionality of the original data tables to
a few principal axes of maximum co-variance. While, e.g., two-
dimensional projection is ideal for visual inspection, the corresponding data
will be very small so that any reverse engineering GRN method. We therefore back-transform the CIA results,
yielding tables of the original format whose content is solely based on the
selected eigenvectors (Dray and Dufour, 2007). We now briefly describe the
underlying mathematical basis of the back-transformation method following
the notation of (Dray and Dufour, 2007). Given a data table $X$ with $n$ rows,
$p$ columns and $nf$ kept axes, the approximated data table can be obtained from
the following equations:

\[ K(A_1)_D^{1/2}A_1^{-1/2}X = KDX \]  

(1)

And with the left multiplication of $KD$ we will have:

\[ KDK(A_1)_D^{1/2}A_1^{-1/2}X = KDX \]  

(2)

\[ K(A_1)_D^{1/2}A_1^{-1/2}X = X \]  

(3)

where $D$ is a vector of row weights with length $n$. $A$ is a diagonal matrix
of eigenvalues with length $n$, $r$ is the rank of the diagram where the
non-zero eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_r > 0$ are stored in the diagonal
matrix $A_{\lambda r}$. $K$ is a data matrix of $n$ rows and $nf$ columns and $A$ is a matrix with the
principal axes of $p$ rows and $nf$ columns. The details for reconstruction of
these data are described by (Dray and Dufour, 2007). The derivation of the
duality diagram concept is also described by (Dolédec and Chessel, 1994;
Dray et al., 2003).

2.6 Dynamic Bayesian networks

A Bayesian Network (BN), also called a ‘probabilistic graphical model’,
is a graphical representation of a model that explains the probabilistic
relationship between variables. Each observed variable corresponds to a
node. Directed edges represent conditional dependencies between nodes.
BNs have become a popular method for modeling gene regulatory
networks, since they are able to represent complex stochastic processes and
allow combinatorial and non-linear relationships among variables of complex
biological systems (Friedman et al., 2000; Hartemink et al., 2002a). DBNs are an extension of BN, able to infer interactions from time-series datasets
rather than steady-state data. They can also handle noisy data to capture the
architecture of regulatory networks from microarray data (Smith et al., 2002;
Yu et al., 2004).

DBN inference was carried out utilizing Banjo (Bayesian Network
Inference with Java Objects). It focuses on score-based structure inference.
For each network structure explored, the parameters of the conditional
probability density distribution are inferred and an overall network’s score
is computed using the Bayesian Dirichlet scoring metric (BDe). In Banjo,
heuristic approaches, such as greedy with random restart or simulated
annealing, are used to search for the highest scoring graph among a set of
networks. The output network will be either the top graph (highest score) or
consensus network. The consensus network is computed based on the N top-
scoring networks by assigning exponentially weighted probabilities to the
individual edges in each of the high-scoring networks, based on the ranking
of each network in the set. The probability of edges being present in the
consensus network is computed using the weighted average approximation
between N highest scoring models. The background for the construction
of the consensus graphs is described by (Hartemink et al., 2002b)
Banjo was run on all the data sets using default parameters (Supplementary
data S7). To identify robust interactions among a set of top-scoring networks,
we used consensus networks. The output was rendered with dot, a graph layout visualization tool by AT&T (http://www.graphviz.org/). Since it is possible to run java from within MATLAB, we ran BANJO release 2.0 in MATLAB. We compared the DBN algorithm performance when each model dataset discretized into three, four and five bins. We obtained regulatory networks close to a ‘true’ network compiled from KEGG and other databases when three categories are selected.

2.7 Evaluation
We assessed sensitivity, specificity and accuracy of our approach by comparing the resulting gene networks to a gold-standard of known interactions. A true positive (TP) was counted as interaction that is present both in an observed and an expected network, a false positive (FP) for any edge that was predicted in the learnt network but does not exist in the expected network, a false negative (FN) as an edge that is present in the expected network but not in the learnt network, and a true negative (TN) when an interaction does not exist in either learnt or expected networks. To construct an expected network, we merged all pathways involved in our gene lists into a new graph containing all nodes and edges. Therefore, the expected network represents comprehensive regulatory paths and physical interactions, accounting for the fact that many KEGG (Kanehisa et al., 2008) pathways embed other pathways. We used Ingenuity Pathway Analysis (IPA; http://www.ingenuity.com) to account for experimental findings reported in a variety of data resources, such as BioGRID, IntAct, MINT, KEGG and others as detailed in Supplementary data Table S3. In the expected network, all edges are supported by at least one published reference or from canonical information stored in the protein interaction databases.

2.8 Significance analysis of network motifs
To uncover the structural design principles of the reversely engineered GRN, we assessed the comprised network motifs. Network motifs are patterns occurring significantly more frequently than at random (Milo et al., 2002) in complex biological networks. A large number of comprised motifs indicate authenticity and robustness. Motif detection was carried out using a so-called rand-esu algorithm (Wernicke, 2006), generating the random networks from the reversely engineered consensus network by a series of edge switching operations as the default randomization model. We searched 10 million random networks to obtain a comparison to the consensus network. The higher the number of randomized networks, the more accurate the results. Significance analysis of the motifs was carried out by comparing the occurrence of a motif in the consensus network to the occurrence of the same motif in the randomized network. Z-scores were calculated as the occurrence of a motif in the consensus network minus its random frequency divided by the standard deviation in random networks. The higher the Z-score, the more significant a motif. P-values correspond to the number of random networks in which the motif occurred more often than in the original network, divided by the total number of random networks.

3 ALGORITHM AND RESULTS
3.1 Algorithm
We combined the above described existing methods in an iterative procedure to estimate the common regulatory network from different species. An overview is given in Figure 1. Starting on a pair of preprocessed datasets A and B of no particular numbers of genes and also differing in the numbers of samples (arrays), we iteratively apply methods described in the ‘Methods’ section (CIA, Hungarian matching and k-means clustering) in the following manner (Algorithm 1).

Fig. 1. Overview.

Algorithm 1 Pseudocode.

Input: Preprocessed tables A and B of microarray data, differing in numbers of genes (rows) and samples (columns)
Initialization:
Cluster each table (A and B) into same small no. of gene clusters e.g. 3
Represent each cluster by its centroid
Affiliate each cluster centroid of A to a cluster centroid of B yielding a pairing for each possible pairing do
Use these gene cluster pairs as connecting variables for a CIA of samples to identify the pairing with highest co-inertia
end for
Iteration:
while number of clusters in no. of samples do
while RV increases or remains constant do
Call doMatching ( samples, connecting variables )
end while
Call doMatching ( genes, connecting variables )
end while
while increase number of clusters by 1 do
end while

doMatching: Compute weight matrix (objects X objects) containing penalties for high distances
Use Hungarian algorithm to compute optimal matching between objects
Cluster each data set into n clusters
Affiliate cluster centroids by majority voting of object matches
Use these pairs as connecting variables for the next CIA
Return: RV, connecting variables
Refinement:
Decide: define the number of clusters to be matched based on silhouette values
Rearrangement: Result doMatching for n gene clusters out of decision module, obtaining m paired gene cluster centroids
→ back-transformations → reverse engineering GRN
→ verification of common model
3.1.1 Initialization As an initial step, A and B are (separately) divided into n gene clusters, each. For the results presented, we initialized with n = 3.

Each cluster is represented by its centroid (weighted average) as if it were only one gene representing a typical transcription profile for this cluster. Each cluster centroid of A is paired with one cluster centroid of B. There are n! possible ways to combine A and B, each of which is subjected to CIA to determine the one of highest co-inertia. This affiliation, albeit of low granularity (only three connections), is used as a starting point for iteration.

3.1.2 Iteration The remaining procedure consists of two consecutive parts that are iterated with increasing n (until n reaches the number of samples). Both parts are identical in that they take as input an existing CIA, using its projected distances as weight matrix for Hungarian matching and let the resulting matches vote for cluster affiliations that are in turn basis for the next CIA. However, the two parts differ in that the first part starts on an affiliation of sample clusters in order to improve affiliation of gene clusters and vice versa. This is implemented by calling ‘doMatching’ subroutine.

The first part uses the previously performed CIA, collecting the projected distances between the samples of A and B into a sample(A) × sample(B) matrix which is then subjected to the Hungarian algorithm as a penalty matrix. The resulting matching preferentially pairs samples of low distance (resembling co-ordination). Subsequently, samples of A and B are separately clustered into n sample clusters. Each sample cluster is represented by its cluster centroid (typical sample) and each cluster centroid of A is paired to a cluster centroid of B. The pairs are determined by majority voting of above matches, i.e. any two clusters with the highest number of connections between the comprised samples (arrays) become paired. The paired sample cluster centroids serve as connecting variables for a CIA projecting the genes.

The second part uses these projected distances between the genes of A and B, collecting them into a gene(A) × gene(B) matrix which is then subjected to the Hungarian algorithm as penalty matrix. All operations of the second part resemble those in the first part, but the role of genes and samples are switched. In practice, the second part can be performed after transposing both A and B. Please note that the two parts are consecutively iterated until the co-inertia stops increasing (inner loop) before increasing n. The algorithm terminates as n approaches the number of samples (arrays) of the smaller data set.

3.1.3 Refinement The motivation for this step is to allow a larger n (exceeding the number of samples) for the genes. ‘Refinement’ consists of two modules, ‘Decider’ and ‘Rearrangement’. The ‘Decider’ determines whether the number of clusters proposed by the iteration part is accepted as the optimum or if there is room for improvement by further increasing n for the genes. The choice of the decider is tightly connected to the Silhouette values (Rousseeuw, 1987) of gene clusters. If a larger n (maybe even larger than the number of samples) improves clustering, ‘Decider’ will proceed to determine the optimum number of clusters. Subsequently, ‘Rearrangement’ generates m pairs of gene cluster centroids by calling the ‘doMatching’ subroutine.

3.1.4 Reverse Engineering gene regulatory networks For the resulting gene cluster centroids, the CIA coordinates are back-transformed into a data table. Its format and scale resemble that of a conventional microarray data table but it comprises only the variance that is common to both input data tables (of gene cluster centroids). It is subjected to DBN inference resulting in a graph each node of which represents a (cross-species) pair of gene clusters, while its edges stand for interdependencies detected for both species.

Back-transformation, DBN, as well as motif analysis are not used as parts of the algorithm. Apart from that revealing the underlying common gene regulatory network can be rewarding in and of itself, the resulting networks serve as a means to validate our algorithm.

3.2 Cell cycle data—cerevisiae versus pombe

To introduce the method, we show its performance on two closely related datasets, one of which is tailored to resemble the other for another species. Spellman and coworkers recorded mRNA levels for 6178 open reading frames (ORFs) of Saccharomyces cerevisiae over two cell cycle periods in a yeast culture synchronized initially in the cell cycle stage M/G1 at 7-minute intervals for 119 min. Rustici and coworkers monitored mRNAs whose levels oscillate during the cell cycle for 6978 ORFs of Schizosaccharomyces pombe as a function of time in cells synchronized through centrifugal elutriation for 285 min and temperature-sensitive cell cycle mutants for 270 min at 15 min intervals. Both datasets were recorded on glass-slides using two-channel fluorescent labeling. Generally, synchronization substantially decreases after two periods. In order to maximize similarity, we selected 10 time points of highest synchronization and quality from either dataset. We will refer to these data as ‘Sce’ and ‘Spo’, respectively.

The algorithm succeeded in producing the correct matching of time points after 20 iterations. Challenging the ability of our algorithm to reconstruct the correct order of time points without any knowledge about affiliation of neither time points nor gene orthologs, we randomly permuted the sequence of both the time points and genes. Typically, after 16–35 iterations the algorithm converged to the very same result (data not shown).

In the shown example, the algorithm terminated with an RV coefficient of 0.8983. While the algorithm’s outer loops improved the matching score with increasing granularity, the inner loops optimized overall co-structure for a given n (Supplementary Fig. S4). The algorithm gradually increased the matching score in minimum two consecutive inner loops and identified the best similarity score by finding the correct affiliations of the connecting variables in seven outer loops. The result was verified in terms of optimal co-inertia and granularity as detailed in section S6 of supplementary data. The result was visualized by CIA (Supplementary Fig. S2).

Here, the two pairs of projection coordinates are highly correlated and the overall similarity in the structure of the dataset was very high resulting in an RV coefficient of 0.8983. Clearly, the algorithm was able to detect and highlight the similarity between histograms in these datasets, projecting them all in a cluster of histones differentiated from other functionally related genes (Supplementary Fig. S2a, encircled in black).

Table 1 shows that also the other affiliated clusters comprise common functionalities and orthologs. We performed GO term enrichment analysis (Huang da et al., 2009) and listed significant common terms along with the percentage of the involved genes in each cluster. Top common functions are represented by significant
## Table 1. Characterization of the affiliated gene clusters

<table>
<thead>
<tr>
<th>Node</th>
<th>Genes</th>
<th>Orth. Count</th>
<th>Category</th>
<th>Top-common over-expressed biological functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>g1 (7)</td>
<td>64 65</td>
<td>88% 91%</td>
<td>GO:0006996, organelle organization and biogenesis</td>
<td>23 22 35.94% 34.92% 2.171E-02 1.591E-02</td>
</tr>
<tr>
<td>g1</td>
<td>7</td>
<td>64 63</td>
<td>91%</td>
<td>88% 35.94% 34.92% 2.171E-02 1.591E-02</td>
</tr>
<tr>
<td>g1</td>
<td>7</td>
<td>64 63</td>
<td>91%</td>
<td>88% 35.94% 34.92% 2.171E-02 1.591E-02</td>
</tr>
<tr>
<td>g11 (3, G2)</td>
<td>17 16</td>
<td>81% 94%</td>
<td>cell cycle</td>
<td>3 3 17.65% 23.08% 6.271E-04 2.609E-02</td>
</tr>
<tr>
<td>g10 (12, G1)</td>
<td>22 27</td>
<td>90% 89%</td>
<td>protein binding</td>
<td>17 9 38.89% 16.67% 7.437E-03 7.172E-03</td>
</tr>
<tr>
<td>g9 (1)</td>
<td>10 10</td>
<td>100% 100%</td>
<td>Histones</td>
<td>9 7 15.55% 8.33% 3.824E-02 2.191E-02</td>
</tr>
<tr>
<td>g8 (11)</td>
<td>13 10</td>
<td>89% 95%</td>
<td>GO:0005933, cellular bud</td>
<td>6 3 10.20% 6.25% 1.274E-02 4.888E-04</td>
</tr>
<tr>
<td>g7 (6, M)</td>
<td>49 32</td>
<td>91% 86%</td>
<td>GO:0006323, DNA packaging</td>
<td>8 11 12.50% 17.46% 2.080E-02 4.230E-04</td>
</tr>
<tr>
<td>g6 (5)</td>
<td>31 41</td>
<td>64% 88%</td>
<td>GO:0000696, cell cycle</td>
<td>23 23 40.74% 53.46% 9.538E-03 7.771E-03</td>
</tr>
<tr>
<td>g4 (10)</td>
<td>36 54</td>
<td>94% 90%</td>
<td>GO:0005794, regulation of cellular process</td>
<td>19 19 65.00% 22.62% 2.472E-02 1.928E-02</td>
</tr>
<tr>
<td>g3 (5)</td>
<td>13 10</td>
<td>89% 95%</td>
<td>GO:0005794, regulation of cellular process</td>
<td>19 19 65.00% 22.62% 2.472E-02 1.928E-02</td>
</tr>
<tr>
<td>g2</td>
<td>12</td>
<td>46 47</td>
<td>87% 90%</td>
<td>GO:0050794, DNA degradation DNA repair</td>
</tr>
<tr>
<td>g1</td>
<td>17</td>
<td>13 11</td>
<td>81% 94%</td>
<td>GO:0005794, cell cycle</td>
</tr>
</tbody>
</table>

# The sequence of the nodes in the cell cycle is provided in brackets along with their cell-cycle affiliations.
# This column shows the number of correctly affiliated orthologs as a percentage of all orthologs "available" for this gene cluster.
# Number of genes known to be involved in the same functional category (GO-term) in each individual gene cluster.
Cross-species meta-analyses without prior gene-affiliation

Fig. 2. Common ‘Sce’ and ‘Spp’ regulatory network. Affiliated gene clusters are represented as nodes, their interactions as edges. These interactions are color-coded according to their occurrence in KEGG or one of the other pathway databases listed in the ‘Methods’ section. True positive (TP) edges are shown in green, missing edges (FN) are shown in black, incorrect or previously unknown interactions (FP) are shown in red. Any green or black edge is supported by at least one publication.

Fig. 3. Network inferred from the single ‘Sce’ dataset. The layout follows Figure 2.

The observed edges can be explained by transcription factor activity. For the direct edge from g11 to g3, transcription factors SIM1 and FKH1 of g11 have been shown to regulate 2 and 8 genes in g3, respectively (genes and literature are provided in Supplementary Table S5). For the path from g11 to g3 via g12, 17 genes of g12 are targets of the transcription factor SFP1. While SFP1 itself was filtered out for showing unreliable small signals, it is regulated by KARS, RPC11 and SMC3 of cluster g11 (Supplementary Table S5). From g12, the comprised transcription factors RDS2 and MAL33 are known to regulate PDR16 and ALG14, OPY2 and RAD353 of g3, respectively (Supplementary Table S5).

While the edge from g11 to g12 can also be obtained from the ‘Sce’ dataset alone (Fig. 3), the edge from g6 to g12 is not present in either single network (Figs 3 and 4) but is only detected by combining the datasets (Fig. 2). The same is true for the above described edge between g9 and g11. The superiority of the common network is quantified in Section 3.3.

Out of 144 possible directed interactions, 53 true positives, 5 false-positives, 36 false-negatives and 50 true-negatives were detected. Assuming that any interaction listed in any database for these genes would be detectable from these small datasets, sensitivity is 60%. Thus, most (more than half) interactions in pathway databases are present in these data, common to both datasets, and successfully detected here, with 72% accuracy and a specificity of 91%. Furthermore, we assessed the coherence of the interactions found, i.e. their tendency to form sound regulatory modules, by network motif analysis. Size-3 and size-4 subgraph frequencies were determined by generating 10 million directed random graphs with same sample probabilities and in which cases the probability that a given edge exists was preserved. For this, we calculated all 13 non-isomorphic directed size-3 subgraphs and all 21 network motifs listed in Supplementary Table S2 exhibit $P < 0.05$ as well as $Z$-scores greater than two.
Although both datasets had been recorded on the generally more
accurate Affymetrix Gene Chip platform and although both comprise
more inertia (information) than for the yeast data (0.207 and 0.2716
as opposed to 0.115 and 0.1393), RV decreases to 0.8. However,
that still warrants considerable co-structure. The resulting common
model (Supplementary Fig. S3) appears as accurate (69%) compared
to the yeast common network (72%).

For the above examples, we were not provided with ortholog
information for all genes. A third pair of datasets provides more
direct evidence in that both datasets stem from the same species,
thus all gene affiliations are known. Both samples and genes were
permuted for one of the two S. pombe cell cycle experiments
described in the Supplementary data Section S8. Our algorithm was
able to reconstruct correct affiliations of all samples (Supplementary
Fig. S10) as well as for 87% of all genes (Supplementary Table S4).

### 4 DISCUSSION

Co-expression has been widely used to reveal, amongst others,
functional relationships (Adie et al., 2006; Lage et al., 2007) or
to identify common regulatory motifs (Brunner and van Driel,
2004; Franke et al., 2006). Much like conserved sequence motifs,
important regulatory patterns can be observed across species
borders. In order to account for different scales such datasets may
have, co-expression can be determined on the basis of intermediate
results such as vote counting (Rhodes et al., 2004; Smid et al.,
2003), probabilities (Tijporkova and Boeva, 2008) or ranks. However,
in order not to lose any information beforehand, we perform
information reduction in the very process of combination. Co-inertia
analysis (Dolédec and Chessel, 1994) is particularly well-suited for
this task, reducing dimensions based on the common variance (co-
inertia) of two datasets. It can deal with datasets whose variables
(genes) far exceed the number of samples (arrays) and its use for
microarray data has been demonstrated before (Culhane et al.,
2003).

As a prerequisite, however, it requires either the samples or
the genes to be affiliated beforehand. In a cross-species survey of
different samples, those genes would have to be reliably affiliated
between datasets. However, sequence similarity based orthology
does not account for evolutionary phenomena such as sub- and
neo-functionalization, thus not necessarily representing functional
orthology in every case (Fierro et al., 2008). Here, instead of
identifying orthologs beforehand, affiliations are computed by our
algorithm on the basis of the expression data.

In an approach solely based on co-expression, genes that show
identical expression behavior are indistinguishable, thus becoming
one single entity. This entity can be viewed as a node in a GRN.
Comparing such networks with known interactions supplied by
KEGG and other repositories can provide an additional means to
evaluate the performance of our algorithm. To this end, out
of many algorithms proposed for network inference, we picked
DBN as one of the successful algorithms to date for time-series
(Smith et al., 2003; Yu et al., 2004). Non-time series data can be
handled, for example by information-theoretic approaches (Basso
et al., 2005) or algorithms based on ordinary differential equations
(ODE) following transcriptional perturbations (Bansal et al.,
2006). For DBN inference, as for other GRN inference methods, the number
of observations is critical. In general, due to the lack of samples, only
few genes can make it as nodes for stable network inference.

In order to obtain number and composition of nodes optimal for
inferring a common network, our algorithm increases the granularity
step by step (outer loop). For each n, the inner loop pairs the

---

**Table 2. Comparison of each single dataset to the common network in Yeast**

<table>
<thead>
<tr>
<th></th>
<th>“Sce” network</th>
<th>“Spo” network</th>
<th>Common network</th>
</tr>
</thead>
<tbody>
<tr>
<td>True positive edges</td>
<td>17</td>
<td>21</td>
<td>53</td>
</tr>
<tr>
<td>False positive edges</td>
<td>22</td>
<td>25</td>
<td>5</td>
</tr>
<tr>
<td>False negative edges</td>
<td>31</td>
<td>26</td>
<td>36</td>
</tr>
<tr>
<td>True negative edges</td>
<td>74</td>
<td>72</td>
<td>50</td>
</tr>
<tr>
<td>Sensitivity (%)</td>
<td>77</td>
<td>74</td>
<td>91</td>
</tr>
<tr>
<td>Specificity (%)</td>
<td>63</td>
<td>64</td>
<td>72</td>
</tr>
<tr>
<td>Number of network motifs</td>
<td>13</td>
<td>18</td>
<td>21</td>
</tr>
</tbody>
</table>

---

3.3 Superiority of the common network

In order to assess the advantage of combining datasets using our
algorithm, we compared the common network to the networks
obtained from each single dataset. The networks inferred from ‘Sce’
and ‘Spo’ datasets are shown in Figures 3 and 4, respectively. The
specificity and sensitivity of the networks compared to the common
network is summarized in Table 2. The common network improves
upon the single (‘Sce’ and ‘Spo’) networks in terms of absolute
numbers of true positive and false positive edges, as well as in
sensitivity, specificity, accuracy and the number of network motifs
(Table 2).

3.4 Application to further datasets

The first pair of datasets is ideal for verifying correct affiliation
of the time points (Supplementary Fig. S2). One set was made
to resemble the other where possible, extending the cell cycle
transcription studies to another species. In order to demonstrate the
performance of our algorithm in a real-world-scenario, we picked
a second pair of more distantly related datasets modeling the reactions
of mouse and human to estradiol (Supplementary data Section S5).
Although both datasets had been recorded on the generally more
crisp Affymetrix Gene Chip platform and although both comprise

---

**Fig. 4.** Network inferred from the single ‘Spo’ dataset.
Thus, the chosen granularity, although it is small in comparison to e.g. affiliating two genes (or clusters thereof) may be reversed in the previous one, the approach is non-greedy in that each decision on matching and majority voting, alternating between ‘connecting network motifs all coincide for Silhouette values, for true positive interactions and for the yield of point for our algorithm. The largest (optimal) numbers for RV, for common GRN does not appear less accurate than that for the first. As GRN. Although the second pair of datasets shows less similarity, the remarkably, 60% sensitivity in comparison to known interactions. and the delineated edges show 72% accuracy, 91% specificity and, (arrays) by far if at all. However, after back-transformation and a yielding granularity never exceeded the number of samples part while for the second dataset the decider module determined that further refinement was not beneficial. Generally, in our hands a yielding granularity never exceeded the number of samples (arrays) by far at all. However, after back-transformation and RE, the inferred network comprises 21 significant network motifs and the delineated edges show 72% accuracy, 91% specificity and, remarkably, 60% sensitivity in comparison to known networks. Thus, the chosen granularity, although it is small in comparison to the number of genes, resulted in a robust and most informative network.

Furthermore, the common network shows increased specificity, sensitivity, and accuracy, as well as more significant network motifs if compared to the networks inferred from the single datasets. This demonstrates that it is possible to successfully combine datasets solely on the basis of co-expression, without applying any further information. To our knowledge, our algorithm represents a novelty in this respect.

External knowledge can be made available to the method via the penalty matrices. These can be weighted according to known similarities between genes and/or between samples. Here, however, we choose to use all external knowledge for evaluation purposes. Requiring no beforehand affiliation, our algorithm can be used for automated large-scale combination of microarray datasets. Back-transformation results in an artificial data table containing only the variance common to the two initial tables while retaining the scale of the first table. Thus, it can be handled like any real data table, e.g. for subsequent GRN inference or for combining it with yet another real data table or a combination of such. Thus, the method can be extended to linking more than two datasets, either hierarchically merging back-transformed data tables, or by using multiple co-inertia analysis.

With increasing numbers of datasets to be summarized in one model, the common variance will decrease. Generally speaking, we would expect a tendency for such a model to be small, widely applicable, robust, and relatively free of noise and systematic errors when multiple experimental platforms are mixed. Extensive cross-species models could be useful in a pharmacological context in order to predict if a model organism closely resembles a human regulatory mechanism to interfere with.

Furthermore, the application of our algorithm is not limited to microarray data. It could serve to integrate proteomic, transcriptomic and high-throughput methylation data recorded for the same samples.

ACKNOWLEDGEMENTS

We thank Jörg D. Hoheisel, Christoph Schröder, David Ittai Zhang, Yasser Riazalhosseini, Jorge Sozaried and Rafael Queiroz for helpful discussions, and Lisa Mullan and Bernhard Küster for critical reading of the article. We thank the anonymous reviewers for helpful suggestions.

Funding: NGFN program of the German Federal Ministry of Education and Research; Support for the development of the software platform ‘M-CHIPS’ was provided by the Helmholtz Association.

Conflict of Interest: None declared.

REFERENCES


