BIOINFORMATICS ORIGINAL PAPER

Vol. 26 no. 8 2010, pages 1082–1090
doi:10.1093/bioinformatics/btq096

Systems biology

Advance Access publication March 2, 2010

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

Amin Moghaddas Gholami[1,2] and Kurt Fellenberg[1,*]

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

Associate Editor: John Quackenbush

ABSTRACT

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

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.

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

Contact: kurt@tum.de

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

*To whom correspondence should be addressed.

nearly identical experimental conditions but on two different species (*Saccharomyces cerevisiae, Schizosaccharomyces pombe*). Furthermore, we applied our algorithm to estrogen-regulated gene expression studies of *Homo sapiens* and *Mus musculus* (Supplementary data S5), thus supporting its general applicability to different levels of similarity, underlying structure and experimental platform (both two-channel cDNA and Affymetrix chips). As a means to verify each and every gene affiliation, we also used a pair of datasets stemming from the same species (Supplementary data S8). Each dataset was pre-processed separately (Supplementary data S1).

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 correlation (Robert and Escoufier, 1976). In CIA, the RV is calculated as the co-inertia (sum of eigenvalues of a co-inertia analysis) divided by the square root of the product of the square inertias (sum of the eigenvalues) from the individual CA (Culhane *et al.*, 2003). Much like a correlation coefficient, the stronger the joint trends between two datasets agree, the closer to 1 the RV score becomes. A zero RV score indicates no similarity.

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 will 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 of 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 homogenous gene or sample (array) clusters by the k-means algorithm. Here, we performed it on the χ^2-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 an, e.g. two-dimensional projection is ideal for visual inspection, the corresponding data table of only two rows (or columns) would be too small for 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:

$$K\Lambda_{[r]}{}^{1/2}A^t = KK^tDX \tag{1}$$

And with the left multiplication of K^tD we will have:

$$K^tDK\Lambda_{[r]}{}^{1/2}A^t = K^tDX \tag{2}$$

$$K\Lambda_{[r]}{}^{1/2}A^t = X \tag{3}$$

where D is a vector of row weights with length n. Λ is a diagonal matrix of eigenvalues with length r.r is called the rank of the diagram where the non-zero eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_r > 0$ are stored in the diagonal matrix $\Lambda_{[r]}$. 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 reconstitution 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 graph 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 among N highest scoring models. The background for the concept 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).

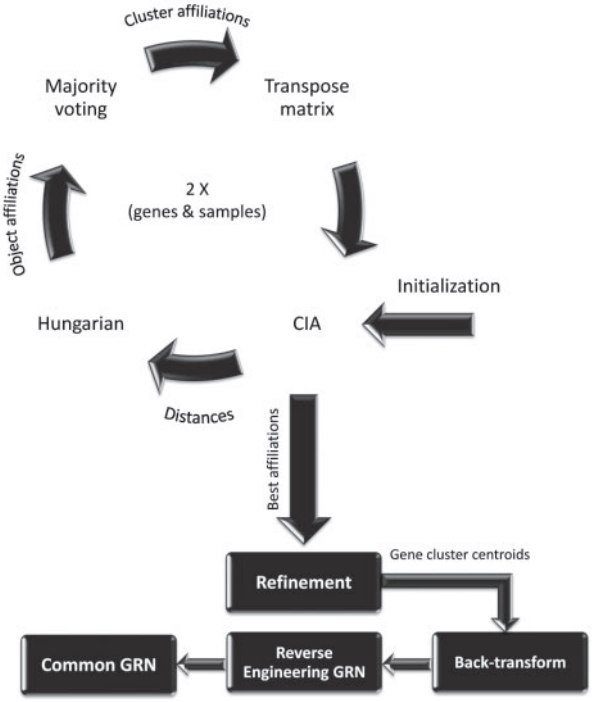

Fig. 1. Overview.

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 *n* (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 <= no. of samples **do**
    while RV increases **or** remains constant **do**
      Call doMatching (samples, connecting variables)
      Call doMatching (genes, connecting variables)
    end while
    increase number of clusters *n* by 1
  end while

doMatching:
  Compute weight matrix (objects × 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:
  Decider: define the number of clusters to be matched based on *silhouette values*
  Rearrangement: Recall **doMatching** for *m* gene clusters out of decider module, obtaining *m* paired gene cluster centroids

→ back-transformation → 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 inter-dependencies 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 a RV coefficient of 0.8983. Clearly, the algorithm was able to detect and highlight the similarity between histones 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

Node[a]	Genes		Orth. Counts[b]		Top common over-represented biological functions						
	'Spo'	'Sce'	'Spo'	'Sce'	Category	Spo[c]	Sce[c]	Spo %	Sce %	Spo pvalue	Sce pvalue
g1 (7)	64	63	88%	91%	GO:0006996_organelle organization and biogenesis	23	22	35.94%	34.92%	2.171E-02	1.591E-02
					GO:0043228_non-membrane-bound organelle	17	21	26.56%	33.33%	6.318E-03	5.305E-03
					GO:0043232_intracellular non-membrane-bound organelle	17	21	26.56%	33.33%	6.318E-03	5.305E-03
					GO:0051276_chromosome organization and biogenesis	15	15	23.44%	23.81%	4.290E-04	8.990E-05
					GO:0007049_cell cycle	12	16	18.75%	25.40%	8.986E-03	8.986E-03
					GO:0022402_cell cycle process	11	14	17.19%	22.22%	3.872E-02	1.661E-02
					GO:0051321_meiotic cell cycle	8	15	12.50%	23.81%	2.463E-03	8.990E-05
					GO:0022403_cell cycle phase	10	12	15.63%	19.05%	3.487E-02	1.573E-02
					GO:0044427_chromosomal part	8	14	12.50%	22.22%	1.230E-02	4.290E-04
					DNA binding	15	16	23.44%	25.40%	1.817E-03	3.598E-02
					GO:0000279_M phase	9	12	14.06%	19.05%	1.425E-02	8.800E-03
					GO:0006323_DNA packaging	8	11	12.50%	17.46%	2.080E-02	4.230E-04
g2 (9,M/G1)	54	43	85%	97%	GO:0016043_cellular component organization	23	23	42.59%	53.49%	5.243E-03	2.720E-03
					GO:0043283_biopolymer metabolic process	22	23	40.74%	53.49%	9.538E-03	7.773E-03
					GO:0032553_ribonucleotide binding	10	14	18.52%	32.56%	6.023E-03	9.034E-03
					GO:0007049_cell cycle	11	5	20.37%	11.63%	7.773E-03	5.459E-03
					DNA binding	5	10	9.26%	23.26%	3.772E-02	3.772E-02
					GO:0000278_mitotic cell cycle	7	6	12.96%	13.95%	3.694E-02	3.694E-02
					GO:0000087_M phase of mitotic cell cycle	6	8	11.11%	18.60%	5.459E-03	4.214E-02
g3 (8)	60	84	68%	84%	GO:0005515_protein binding	39	19	65.00%	22.62%	2.472E-02	1.927E-03
					GO:0065007_biological regulation	20	24	33.33%	28.57%	8.096E-03	1.846E-03
					GO:0000082_G1/S transition of mitotic cell cycle	9	7	15.00%	8.33%	3.370E-02	5.871E-03
					GO:0032502_developmental process	8	13	13.33%	15.48%	3.987E-02	1.264E-02
					GO:0007049_cell cycle	15	5	25.00%	5.95%	6.160E-03	3.343E-02
					GO:0051301_cell division	10	10	16.67%	11.90%	8.302E-03	1.056E-02
					GO:0030427_site of polarized growth	11	7	18.33%	8.33%	4.623E-05	4.252E-05
					GO:0000074_regulation of progression through cell cycle	7	8	11.67%	9.52%	4.317E-02	1.969E-02
					GO:0005856_cytoskeleton	7	8	11.67%	9.52%	1.969E-02	7.652E-03
					GO:0005933_cellular bud	10	5	16.67%	5.95%	5.739E-03	2.264E-04
					GO:0048519_negative regulation of biological process	9	6	15.00%	7.14%	5.739E-03	3.370E-02
					GO:0051726_regulation of cell cycle	7	8	11.67%	9.52%	4.317E-02	1.969E-02
g4 (10)	36	54	94%	90%	GO:0050789_regulation of biological process	14	9	38.89%	16.67%	8.656E-03	7.954E-03
					GO:0050794_regulation of cellular process	14	9	38.89%	16.67%	7.434E-04	7.123E-03
					GO:0065007_biological regulation	16	5	44.44%	9.26%	1.895E-03	1.895E-03
					GO:0022402_cell cycle process	9	7	25.00%	12.96%	9.051E-04	7.317E-03
					GO:0000278_mitotic cell cycle	8	6	22.22%	11.11%	1.852E-02	1.625E-02
					GO:0007088_regulation of S phase	2	3	5.56%	5.56%	7.269E-04	2.535E-02
g5 (6, M)	49	32	91%	86%	GO:0051301_cell division	6	9	12.24%	28.13%	3.110E-02	1.533E-02
					GO:0000278_mitotic cell cycle	5	7	10.20%	21.88%	9.787E-04	4.132E-02
					GO:0000279_M phase	8	3	16.33%	9.38%	3.110E-02	7.395E-03
					GO:0044430_cytoskeletal part	4	7	8.16%	21.88%	9.895E-03	8.687E-03
					cell cycle control	3	6	6.12%	18.75%	1.274E-02	4.888E-03
g6 (5)	31	41	64%	88%	transmembrane protein	14	10	45.16%	24.39%	5.725E-03	2.625E-02
					GO:0051301_cell division	6	6	19.35%	14.63%	9.339E-03	4.124E-02
					GO:0007010_cytoskeleton organization	5	5	16.13%	12.20%	6.161E-04	3.497E-02
g7 (11)	13	10	89%	95%	GO:0000278_mitotic cell cycle	5	5	38.46%	50.00%	7.088E-03	1.828E-03
					cell cycle	4	4	30.77%	40.00%	1.569E-02	4.554E-03
					GO:0051301_cell division	4	4	30.77%	40.00%	3.522E-02	2.428E-02
g8 (1, S)	10	10	100%	100%	**Histones**						
g9 (2)	13	21	75%	83%	GO:0016043_cellular component organization	9	13	69.23%	61.90%	8.489E-04	6.756E-03
					cell cycle	5	6	38.46%	28.57%	1.027E-02	2.980E-03
g10 (12, G1)	22	27	90%	89%	GO:0005515_protein binding	17	8	77.27%	29.63%	3.957E-02	5.390E-03
					GO:0006996_cell cycle	6	9	27.27%	33.33%	8.582E-03	6.943E-03
					DNA damage	3	5	13.64%	18.52%	4.905E-02	3.401E-02
					GO:0006261_DNA-dependent DNA replication	3	4	13.64%	14.81%	5.390E-03	1.687E-02
g11 (3, G2)	17	13	81%	94%	cell cycle	3	3	17.65%	23.08%	6.271E-04	2.609E-02
					GO:0007047_cell wall organization and biogenesis	3	3	17.65%	23.08%	6.892E-03	2.588E-02
g12 (4)	24	46	87%	100%	GO:0016043_cellular component organization	12	26	50.00%	56.52%	7.028E-05	2.741E-02
					GO:0065007_biological regulation	8	19	33.33%	41.30%	9.026E-04	4.175E-03
					phosphoprotein	3	17	12.50%	36.96%	9.403E-03	5.889E-03
					GO:0032502_developmental process	4	10	16.67%	21.74%	8.368E-02	1.386E-02

[a]The sequence of the nodes in the cell cycle is provided in brackets along with their cell-cycle affiliations.
[b]This column shows the number of correctly affiliated orthologues as a percentage of all orthologues "available" for this gene cluster.
[c]Number of genes known to be involved in the same functional category (GO-term) in each individual gene cluster.

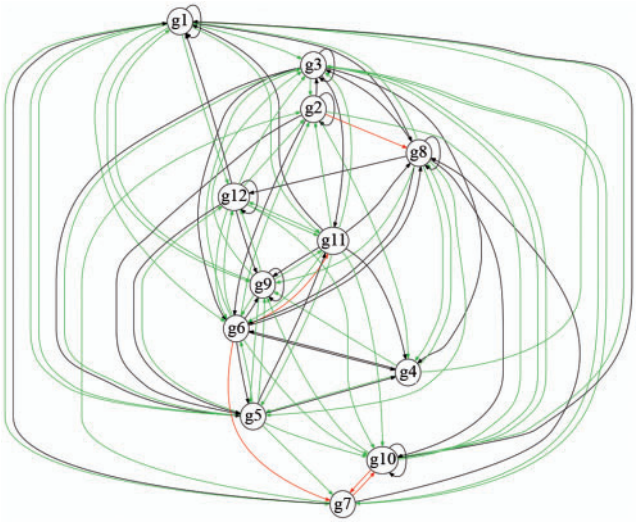

Fig. 2. Common '*Sce*' and '*Spo*' regulatory network. Affiliated gene clusters are represented as nodes, their interactions as edges. These interactions are color-coded according to their occurrence in KEGG or one of the other pathway databases listed in the 'Methods' section. True positive (TP) edges 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.

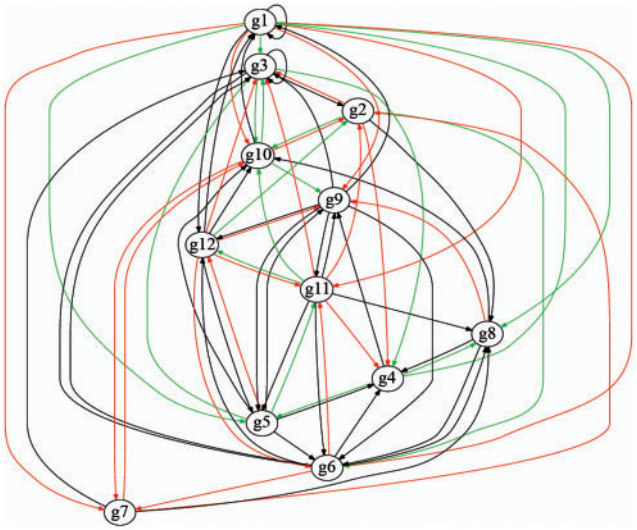

Fig. 3. Network inferred from the single '*Sce*' dataset. The layout follows Figure 2.

associations of P-value <0.05. In the same manner, we summarized the percentages of correctly affiliated orthologs. A complete table listing all genes is given in the Supplementary Table S1. Based on the two back-transformed data tables, we represented each cluster by the gene-wise sum of all comprised genes across both tables. Subjecting this combined table to DBN algorithm, these cluster representatives became the nodes of a common gene network.

A graphical representation of the resulting common network is shown in Figure 2. To illustrate it by example, we follow its edges from the smallest to the largest cluster, moving through the cell cycle from S phase towards mitosis (Table 1). The histones (cluster g8) play an important role in transcriptional regulation. We observe an edge from g8 to g9 comprising the cyclin *CLN2* comparing favorably with work of Santisteban and coworkers (Santisteban *et al.*, 1997).

Following the cell cycle from S to G2, the cohesin complex is required to hold together the sister chromatids. This process is mediated by the acetyltransferase *ECO1* of cluster g9 (S-phase) directly interacting with cohesion complex subunit *SMC3* (G2-phase) of cluster g11 (Ben-Shahar *et al.*, 2008; Rowland *et al.*, 2009; Unal *et al.*, 2008). The edge linking g9 and g11 suggests an according tight transcriptional regulation of acetylase *ECO1* preceding *SMC3*.

Following the cell cycle from G2 to M, the common transcription network shows node g11 (G2) to regulate both g12 and g3, whereas g12 itself also regulates g3, forming a network motif referred to as feedforward loop (Alon, 2007). It is often found in the context of signal amplification. The increase in cellular activity during G2/M transition is also reflected by increased glycolysis (*GLK1, PFK1*) and by g3 being the largest cluster. While still transcribing genes important for G2/M transition (*SWE1*) and DNA repair (*RAD53, EXO1, MSH2, POL3*), the cell already prepares for budding (*BEM1, SPH1, FAA1, BNI5, GIC1*) and cytokinesis (*HOF1, MYO3, STU2, YPT11*).

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 unreliably small signals, it is regulated by *KAR5, 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 *RAD53* 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 as well as 199 non-isomorphic directed size-4 sub graphs. All 21 network motifs listed in Supplementary Table S2 exhibit $P < 0.05$ as well as Z-scores greater than two.

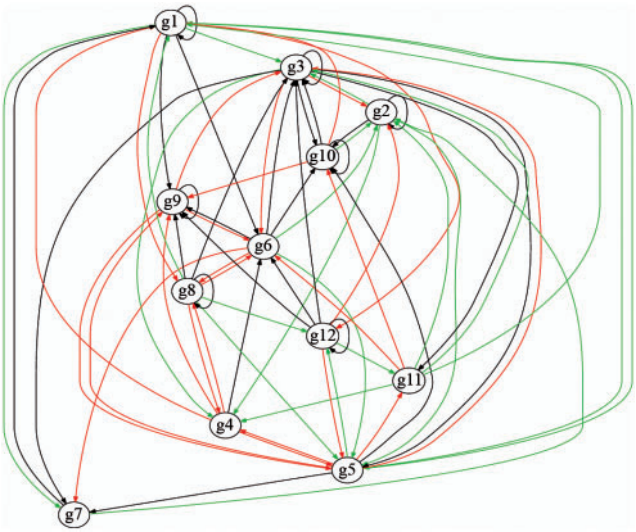

Fig. 4. Network inferred from the single '*Spo*' dataset.

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

	"*Sce*" network	"*Spo*" network	Common network
True positive edges	17	21	53
False positive edges	22	25	5
False negative edges	31	26	36
True negative edges	74	72	50
Sensitivity (%)	35	44	60
Specificity (%)	77	74	91
Accuracy (%)	63	64	72
Number of network motifs	13	18	21

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 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 (Tsiporkova 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

n clusters of each dataset, seeking for an inter-datasets affiliation of optimal co-inertia. It does so via a combination of CIA, Hungarian matching and majority voting, alternating between 'connecting variable' affiliations. While each of these steps 'learns' from the previous one, the approach is non-greedy in that each decision on an e.g. affiliating two genes (or clusters thereof) may be reversed in the next iteration.

Gene cluster affiliations were evaluated directly (within the same species), by counting gene orthologs, and by inferring common GRN. Although the second pair of datasets shows less similarity, the common GRN does not appear less accurate than that for the first. As the terminal granularity (final number of distinguishable clusters) is crucial for network inference, we carefully evaluated the termination point for our algorithm. The largest (optimal) numbers for RV, for Silhouette values, for true positive interactions and for the yield of network motifs all coincide for $n = 12$ in the first example as well as for $n = 10$ in the second. In the first example, the iteration part of the algorithm could not come up to the final n because the number of clusters cannot exceed the number of samples in the smaller dataset (here both 10). Instead, $n = 12$ was determined by the refinement 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 if 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 interactions. 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 Jitao 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

Adie,E.A. *et al.* (2006) SUSPECTS: enabling fast and effective prioritization of positional candidates. *Bioinformatics*, **22**, 773–774.

Alon,U. (2007) Network motifs: theory and experimental approaches. *Nat. Rev. Genet.*, **8**, 450–461.

Bansal,M. *et al.* (2006) Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, **22**, 815–822.

Basso,K. *et al.* (2005) Reverse engineering of regulatory networks in human B cells. *Nat. Genet.*, **37**, 382–390.

Ben-Shahar,T.R. *et al.* (2008) Eco1-dependent cohesin acetylation during establishment of sister chromatid cohesion. *Science*, **321**, 563–566.

Bourgeois,F. and Lassalle,J.-C. (1971) An extension of the Munkres algorithm for the assignment problem to rectangular matrices. *Commun. ACM*, **14**, 802–804.

Brunner,H.G. and van Driel,M.A. (2004) From syndrome families to functional genomics. *Nat. Rev. Genet.*, **5**, 545–551.

Choi,J.K. *et al.* (2003) Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics*, **19** (Suppl. 1), i84–i90.

Conlon,E.M. *et al.* (2007) Bayesian meta-analysis models for microarray data: a comparative study. *BMC Bioinformatics*, **8**, 80.

Culhane,A.C. *et al.* (2003) Cross-platform comparison and visualisation of gene expression data using co-inertia analysis. *BMC Bioinformatics*, **4**, 59.

Dolédec,S. and Chessel,D. (1994) Co-inertia analysis: an alternative method for studying species-environment relationships. *Freshwater Biol.*, **31**, 277–294.

Dray,S. *et al.* (2003) Co-inertia analysis and the linking of ecological data tables. *Ecology*, **84**, 3078–3089.

Dray,S. and Dufour,A.-B. (2007) The ade4 Package: implementing the duality diagram for ecologists. *J. Stat. Soft.*, **22**, 1–20.

Fellenberg,K. *et al.* (2001) Correspondence analysis applied to microarray data. *Proc. Natl Acad. Sci. USA*, **98**, 10781–10786.

Fierro,A.C. *et al.* (2008) Meta analysis of gene expression data within and across species. *Curr. Genomics*, **9**, 525–534.

Franke,L. *et al.* (2006) Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *Am. J. Hum. Genet.*, **78**, 1011–1025.

Friedman,N. *et al.* (2000) Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.

Garrett-Mayer,E. *et al.* (2008) Cross-study validation and combined analysis of gene expression microarray data. *Biostatistics*, **9**, 333–354.

Gilks,W.R. *et al.* (2005) Fusing microarray experiments with multivariate regression. *Bioinformatics*, **21** (Suppl. 2), ii137–ii143.

Hartemink,A. *et al.* (2002a) Bayesian methods for elucidating genetic regulatory networks. *IEEE Intel. Systems*, **17**, 37–43.

Hartemink,A. *et al.* (2002b) Combining location and expression data for principled discovery of genetic regulatory networks. In Altman,R. and Dunker,A.K. (eds), *Pacific Symposium on Biocomputing 2002 (PSB02)*. World Scientific: New Jersey, 437–449.

Huang da,W. *et al.* (2009) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protocols*, **4**, 44–57.

Kanehisa,M. *et al.* (2008) KEGG for linking genomes to life and the environment. *Nucleic Acids Res.*, **36**, D480–D484.

Kuhn,H.W. (1955) The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.*, **2**, 83–87.

Lage,K. *et al.* (2007) A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nat. Biotechnol.*, **25**, 309–316.

Ma,S. and Huang,J. (2009) Regularized gene selection in cancer microarray meta-analysis. *BMC Bioinformatics*, **10**, 1.

Marot,G. *et al.* (2009) Moderated effect size and p-value combinations for microarray meta-analyses. *Bioinformatics*, **25**, 2692–2699.

Milo,R. *et al.* (2002) Network motifs: simple building blocks of complex networks. *Science*, **298**, 824–827.

Quackenbush,J. (2006) Microarray analysis and tumor classification. *N. Engl. J. Med.*, **354**, 2463–2472.

Ramasamy,A. *et al.* (2008) Key issues in conducting a meta-analysis of gene expression microarray datasets. *PLoS Med.*, **5**, e184.

Rhodes,D.R. *et al.* (2002) Meta-analysis of microarrays: interstudy validation of gene expression profiles reveals pathway dysregulation in prostate cancer. *Cancer Res.*, **62**, 4427–4433.

Rhodes,D.R. *et al.* (2004) Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression. *Proc. Natl Acad. Sci. USA*, **101**, 9309–9314.

Robert,P. and Escoufier,Y. (1976) A unifying tool for linear multivariate statistical methods: the RV-coefficient. *Appl. Statist.*, **25**, 257–265.

Rousseeuw, P. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, **20**, 53–65.

Rowland,B.D. *et al.* (2009) Building sister chromatid cohesion: smc3 acetylation counteracts an antiestablishment activity. *Mol. Cell*, **33**, 763–774.

Rustici,G. *et al.* (2004) Periodic gene expression program of the fission yeast cell cycle. *Nat. Genet.*, **36**, 809–817.

Santisteban,M.S. *et al.* (1997) Histone octamer function in vivo: mutations in the dimer-tetramer interfaces disrupt both gene activation and repression. *EMBO J.*, **16**, 2493–2506.

Smid,M. *et al.* (2003) Venn Mapping: clustering of heterologous microarray data based on the number of co-occurring differentially expressed genes. *Bioinformatics*, **19**, 2065–2071.

Smith,V.A. *et al.* (2002) Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics*, **18** (Suppl. 1), S216–S224.

Smith,V.A. *et al.* (2003) Influence of network topology and data collection on network inference. *Pac. Symp. Biocomput.*, **8**, 164–175.

Spellman,P.T. *et al.* (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.

Stevens,J.R. and Doerge,R.W. (2005) Combining Affymetrix microarray results. *BMC Bioinformatics*, **6**, 57.

Tsiporkova,E. and Boeva,V. (2008) Fusing time series expression data through hybrid aggregation and hierarchical merge. *Bioinformatics*, **24**, i63–i69.

Unal,E. *et al.* (2008) A molecular determinant for the establishment of sister chromatid cohesion. *Science*, **321**, 566–569.

Wang,J. *et al.* (2004) Differences in gene expression between B-cell chronic lymphocytic leukemia and normal B cells: a meta-analysis of three microarray studies. *Bioinformatics*, **20**, 3166–3178.

Wernicke,S. (2006) Efficient detection of network motifs. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **3**, 347–359.

Yang,X. and Sun,X. (2007) Meta-analysis of several gene lists for distinct types of cancer: a simple way to reveal common prognostic markers. *BMC Bioinformatics*, **8**, 118.

Yu,J. *et al.* (2004) Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, **20**, 3594–3603.