Consensus and Meta-analysis Regulatory Networks for Combining Multiple Microarray Gene Expression Datasets

Emma Steele and Allan Tucker

Centre for Intelligent Data Analysis
Brunel University

Corresponding author:
Emma Steele
Department of Information Systems and Computing
Brunel University
Kingston Lane
Uxbridge Middlesex UB8 3PH
UK
e-mail: emma.steele@brunel.ac.uk
fax: +44 (0)1895 251686

Abstract

Microarray data is a key source of experimental data for modelling gene regulatory interactions from expression levels. With the rapid increase of publicly available microarray data comes the opportunity to produce regulatory network models based on multiple datasets. Such models are potentially more robust with greater confidence, and place less reliance on a single dataset. However, combining datasets directly can be difficult as experiments are often conducted on different microarray platforms, and in different laboratories leading to inherent biases in the data that are not always removed through pre-processing such as normalisation. In this paper we compare two frameworks for combining microarray datasets to model regulatory networks: *pre-* and *post-learning aggregation.* In pre-learning approaches, such as using simple scale normalisation prior to the concatenation of datasets, a model is learnt from a combined dataset, whilst in post-learning aggregation individual models are learnt from each dataset and the models are combined. We present two novel approaches for post-learning aggregation, each based on aggregating high-level features of Bayesian network models that have been generated from different microarray expression datasets. *Meta-analysis Bayesian networks* are based on combining statistical confidences attached to network edges whilst *Consensus Bayesian networks* identify consistent network features across all datasets. We apply both approaches to multiple datasets from synthetic and real (E. coli and yeast) networks and demonstrate that both methods can improve on networks learnt from a single dataset or an aggregated dataset formed using a standard scale normalisation.

Keywords: Bayesian networks, gene regulatory networks, consensus algorithms, meta-analysis, microarray gene expression data

1 Introduction

Microarrays are the major source of data for gene expression levels, allowing the expression of thousands of genes to be measured simultaneously. Gene Regulatory Networks (GRNs) describe how the expression level of genes affect the expression of the other genes. Modelling GRNs from expression level data is a topic of great interest in current bioinformatics research [1, 2, 3]. In this paper we seek to take advantage of publicly available gene expression datasets generated by similar biological studies. Drawing together a richer and/or broader collection of data has the potential to produce GRN models that are more robust, have greater confidence and place less reliance on a single dataset.

When learning from multiple datasets, there is a choice for *when* to aggregate knowledge within the datasets. Essentially, there are two alternative approaches, which we refer to as *pre-* and *post-learning aggregation*. In pre-learning aggregation, data is combined prior to learning, and a model is learnt from the combined dataset. In post-learning aggregation individual models are learnt from each dataset, and these are combined after learning. In this paper we compare a simple pre-learning aggregation approach of concatenating datasets after scale normalisation with two novel post-learning aggregation approaches for combining Bayesian network models that have been generated from a number of microarray datasets.

Microarray datasets often come from different platforms [4]. This means that the data can contain different biases and it is difficult, or sometimes impossible, to compare the datasets since measurement units may vary. For example two common expression profiling technologies are cDNA microarrays and oligonucleotide microarrays, which measure gene expression in different ways. Oligonucleotide microarrays give estimates of the absolute value of expression whereas cDNA technology measures relative differences in expression between genes. Additionally, studies come from different laboratories meaning that data is collected with different measurement biases under different atmospheric conditions. Previous research has established that comparing between datasets using standard normalisation techniques is difficult and not straightforward [5, 6, 7]. A post-learning aggregation framework can combine microarray datasets generated by different platforms, research groups and laboratories without requiring normalisation. In this framework, learnt models that are generated from each dataset are aggregated, producing a combined model that represents prominent features which occur in all, or a subset of, the individual dataset models.

Bayesian networks [8] have become a popular method for computational modelling of GRNs from expression data since they are able to represent the network qualitatively (with a network graph) and quantitatively (probability distributions quantify the strength of influences and dependencies between nodes/variables in the network graph) and thus are relatively easy to interpret by non-statisticians (e.g. biologists). We use Bayesian networks to model GRNs in our pre- and post-learning aggregation methods. In post-learning aggregation we combine Bayesian network models generated from each dataset using two different approaches. The first of our methods is a *Consensus* approach that identifies the intersections - that is, common edges - amongst the networks generated from different datasets. Only consistent features and dependencies appear in the final Consensus network, reducing the occurrence of spurious relationships. The second technique is based on *Meta-analysis*, an established field of research for combining the statistical outcomes of medical studies [9]. We use an inverse-variance weighting meta-analysis method to combine statistical confidences that are attached to each network edge.

Whilst comparing and combining microarray expression datasets is a popular topic of research in bioinformatics [10, 11], Wang *et al.* [12] are the first (to our knowledge) to address the issue with regards to modelling GRNs and use a post-learning aggregation framework that combines the models for each dataset into an overall, consistent solution. However, their method is based on linear programming where GRNs are represented using non-linear differential equations. In our work we consider their chosen application of a yeast heat-stress sub-network to evaluate our approaches. Our approaches identify a greater number of documented interactions and are evaluated on a more diverse set of studies obtained from different platforms.

The remainder of the paper is organised as follows. In section 2 we describe the consensus and meta-analysis approaches in more detail. Section 3 details our experimental results on synthetic and real E. coli and Yeast gene expression datasets. Finally, in section 4 we discuss our findings and outline directions for future research.

2 Methods

2.1 Bayesian networks

Bayesian Networks (BNs) are graph-based models of probability distributions that capture properties of conditional independence between variables. A BN consists of two components - a Directed Acyclic Graph (DAG) consisting of edges between nodes that represent variables in the domain, and a set of conditional probability distributions associated with each node. If there is an edge from node A to another node B, then A is said to be a *parent* of B, and B is a *child* or *descendant* of A. The directed edges between nodes indicate the existence of influences and dependencies, the strength of which are quantified by the conditional probabilities. BNs have become a popular method for computational modelling of GRNs from expression data since they are relatively easy to interpret by biologists. The expression level of genes are represented by nodes in the network and influences between genes represented by the directed edges.

We use a score-based search method to learn a BN that represents a GRN from microarray expression data. Since the first research by Friedman *et al.* [1], search-and-score BNs have been used frequently in learning gene networks. This approach performs a search through the space of possible networks and scores each structure. The aim is to identify the network with the maximum score. A variety of search strategies can be used, the simplest being a greedy hill-climb. We use a simulated annealing approach in order to limit local maxima. The search begins with an empty network. At each stage of the search, networks in the current neighbourhood are found by applying operators such as *add edge*, *remove edge* and *reverse edge* to the current network.

We use the *Bayesian Information Criterion* (BIC) for scoring candidate networks. The BIC function is a combination of the model log-likelihood and a penalty term that favours less complex models - as such it is similar to the minimum description length:

$$BIC = -2\ log\ P(\theta|D) +\ k\ log(n)$$

where θ represents the model, D is the data, n is the number of observations (sample size) and k is the number of parameters. $log\ P(\theta|D)$ is the log-likelihood while the term $k\ log(n)$ is a penalty term, which helps to prevent overfitting by biasing towards simpler, less complex models. The BIC is part of a family of Information Criterion scoring functions that take a common formulation but with different penalty terms [13]. For example, the Akaike Information Criterion (AIC) [14] has a penalty term of $2k$ (twice the number of parameters), whereas the BIC has the penalty term $k\ log(n)$ that depends on the number of model parameters but also the number of samples. Since the BIC's penalty term takes the number of samples into account it is more appropriate for dealing with microarray datasets, which commonly contain only a small number of sample points.

It is important to mention that more than one DAG may represent the same set of dependencies amongst variables. A set of such DAGs belong to the same equivalence class. It has been shown that equivalent graphs agree on the same underlying undirected structure, but the direction of some edges may vary [15]. Therefore, the equivalence class of a set of DAGs can be represented using a partially directed acyclic graph (PDAG), where only some edges are directed. Chickering [16] derived an algorithm for constructing the PDAG representing the equivalence class for any DAG; we use this to convert the learnt BNs to their equivalence classes.

Friedman *et al.* [17] devised a method for computing a statistical confidence of features within a BN, based on a well-known statistical method, Efron's Bootstrap [18]. Given a dataset D containing N observations, a new dataset is created by re-sampling N times, with replacement from D. A BN is learnt from the re-sampled dataset. This process is repeated m times, resulting in m learnt BNs. An estimate of confidence for each feature is computed by the proportion of networks (or their equivalence classes) that contain that feature. Friedman *et al.* [1] used this method to calculate statistical confidences for GRN features.

We make use of the bootstrapping method to generate more robust network structures for each microarray dataset. When computing confidence estimates, we define a feature as the existence of an edge between two nodes in the network. The BNs learnt from each resampled dataset are converted to PDAGs in order to ensure equivalence classes are represented, and then confidence estimates for each edge are calculated on this set of PDAGs. Thus, the bootstrapped network has a confidence estimate assigned to each network edge. It is important to note that the edge $i \rightarrow j$ may have a different confidence estimate to the edge $i \leftarrow j$. Where directed edges are present in a PDAG, they contribute only to the confidence estimate for the edge in that direction, whereas undirected edges contribute to the confidence estimate for an edge in both directions.

We can obtain a PDAG from a bootstrapped network by thresholding. If an edge has a confidence above the threshold, it is included in the PDAG (and if edges are found in both directions - e.g. from node $i \rightarrow j$ and $i \leftarrow j$, then the edge is undirected). Thus, if directional dependencies have enough support in the bootstrapping process they will be captured and represented in the final thresholded PDAG. Note that this method of thresholding does present the possibility that the extracted PDAG may not be a PDAG - that is, the network structure could be cyclic. In our experiments, this did not occur. However, if it was the case, the network can be converted to acyclic by undirecting an edge in the cycle. The edge to be undirected can be selected by finding which one has the least support to be directed (that is, it has the smallest difference between the confidences in each direction).

2.2 Scale normalisation of microarray data

Normalisation is the transformation of microarray data to adjust for systematic variations (arising from variation in the technology rather than biological variations). There can be substantial scale differences between microarrays - for example, because of changes in the photomultiplier tube settings of the scanner or for other reasons [19].

Scale-normalisation is a commonly used method for a simple scaling of the log-ratios from a series of arrays so that each array has the same median absolute deviation [20]. Each log ratio is transformed using the following formula:

$$M'_{ij} = \frac{M_{ij} - median_i}{MAD_i}$$

where M_{ij} is the log ratio of the jth gene in the ith array and the median absolute deviation MAD_i is defined as the median of absolute deviations from the median: $MAD_i = median_i\{|M_{ij} - median_i|\}$.

Whilst scale normalisation has the benefit of making the arrays within a dataset comparable, theoretically it also means that arrays between datasets are comparable. Thus, we can use scale-normalisation to combine multiple microarray datasets into one, allowing the generation of a single BN from multiple studies. In practise however, bias and artefacts may still remain in the data after scale normalisation. In our experiments, we use scale-normalisation in pre-learning aggregation, in order to concatenate a number of datasets and we also use it to normalise between arrays within individual datasets for post-learning aggregation. Whilst it is not necessary to perform normalisation for post-learning aggregation, it will allow us to directly compare the two approaches and investigate if scale normalisation is enough to combine datasets, or whether post-learning aggregation can obtain more successful results.

2.3 Combining Bayesian network structures

Our post-learning aggregation approaches are based on combining BN models. This has been addressed in previous research in two main ways - qualitatively and quantitatively, which refer to the focus of combination. Quantitative combination is based on aggregating probability distributions [21], whereas qualitative combination is based on combining the graph structures [22]. We focus on qualitative combination as we are concerned with the dependency structure amongst the genes. Next, we introduce two new methods for the qualitative combination of BNs - Consensus Bayesian networks and Meta-analysis Bayesian networks.

2.3.1 Consensus Bayesian networks

Our Consensus approach (see Fig. 1) is based on the identification of consistencies across a set of networks - edges that appear in all, or a certain proportion of the networks in the set are included in the aggregate Consensus network structure. We use a bootstrapping approach to learn the individual PDAGs with edge confidence estimates for each input dataset. We use thresholding (as described in section 2.1) to obtain a final PDAG from each bootstrapped network. Whilst bootstrapped Bayesian networks and thresholding have been used previously to learn more robust gene regulatory networks [1, 23], we use the thresholded bootstrapped networks as inputs to the Consensus algorithm in order to find consistencies across networks that have been generated from multiple datasets.

The basic Consensus algorithm is fairly simple. Each pair of nodes in each input PDAG is considered in turn and an edge between them in the Consensus network is created if such an edge exists in a proportion of the input PDAGs that exceeds the consensus threshold. Assigning the edge direction is a little more complex. If there is no conflict regarding that edge's direction in the input networks then its direction/undirection remains the same in the consensus network. However, if there is conflict, this introduces some uncertainty regarding the edge direction. In the current implementation of Consensus BNs, if there is a majority in the input networks regarding edge direction, then the edge is assigned the majority direction in the Consensus network. Thus, directed edges with enough support will appear in the Consensus network. If there is no majority then the edge is left as 'unknown direction'. Note that we make a distinction in the Consensus network between edges that are undirected and those that are 'unknown'. An edge that is undirected can be reversed, as in equivalent graphs. However uncertainty exists over the direction of an 'unknown' edge, or whether it can be reversed. We flag up 'unknown' edges in the graphs by using edges that are directed both ways, whereas undirected edges have no arrowheads.

Topological fusion [22] is a similar method for combining the graphical structures of multiple networks using graph union. This means that the final network structure links nodes if they are

```
Consensus Bayesian networks
Input: Set of $n$ individual networks (PDAG representation of their equivalence class), consensus
        threshold (between 0 and 1)
Output: Consensus network

for each pair of nodes i,j do
    if an edge $e_{ij}$ exists between nodes i and j in a proportion of individual networks $\geq$ consensus
    threshold then
        include edge $e_{ij}$ in the Consensus network
    end
    if edge $e_{ij}$ exists in the Consensus network then
        /* Assignment of edge direction                                                    */
        if there is no conflict in the input edge directions then
            Consensus edge $e_{ij}$ is the same direction (whether directed or undirected)
        else
            if There is a majority direction (or undirection) then
                Assign edge $e_{ij}$ in the majority direction
            else
                Assign edge $e_{ij}$ as 'unknown' direction
            end
        end
    end
end
```

Figure 1: Consensus Bayesian Networks algorithm

linked in any of the networks. Since graph union can introduce cycles into the network structure, edge reversal is used. This is where an edge $A \rightarrow B$ is reversed and then edges are added between the parent nodes of A to B, and from the parent nodes of B to A. This maintains the underlying relationships between variables under the principle that it preserves the flow of information. The final fused graph contains all edges (some reversed) and nodes that are in the input DAGs, plus those edges that are introduced from edge reversals.

In previous research we have compared the Consensus algorithm to topological fusion [24]. At the consensus threshold $\frac{1}{n}$ (that corresponds to every edge from each of the n networks appearing in the combined structure), the Consensus approach is equivalent to graph union. However, we have found that the topological fusion network does not do as well as a $\frac{1}{n}$ Consensus network, as it is liable to the inclusion of misdirected edges. The key difference is that the Consensus method represents networks using equivalence classes - so if edges are reversible they are left undirected.

2.3.2 Bayesian network meta-analysis

Meta-analysis refers to a set of statistical methods for combining the result of several studies that address a set of related research hypotheses. Meta-analysis originated in medical statistics [9] but recently has been used to identify highly expressed genes across multiple microarray datasets [11]. In medical statistics, meta-analysis is used to combine outcome measures such as incidence rates (e.g. the rate at which new cases of a disease occur in a population) from multiple medical studies.

We have developed an approach called Bayesian networks meta-analysis[1] (see Fig. 2) that uses the fixed-effects meta-analysis method to combine the statistical confidences for each edge over a set

[1] Bayesian network meta-analysis should not be confused with Bayesian meta-analysis, which involves using Bayesian models to perform the meta-analysis.

```
Bayesian Networks Meta-Analysis
Input: Set of $n$ individual networks with statistical confidences attached to each edge
Output: Meta-analysis network with aggregated statistical confidences attached to each edge

for each edge from node $i \rightarrow j$ (denoted $e_{ij}$) do
    let $T_{ij}(k)$ be the statistical confidence for edge $e_{ij}(k)$ in the $k$th network.
    Calculate the aggregated confidence $\bar{T}$ for edge $e_{ij}$
    using $log(\bar{T}) = \frac{\sum_{k=1}^{n} w_k log(T_{ij}(k))}{\sum_{k=1}^{n} w_k}$
    where $w_k = d_{ij}(k)$, the number of networks learnt during bootstrapping where an edge $i \rightarrow j$ exists
end
```

Figure 2: Bayesian Networks Meta-Analysis algorithm

of bootstrapped networks, producing a single network that has an aggregated confidence attached to each edge. The fixed-effects model assumes no heterogeneity between study results. Whilst this is obviously a naïve assumption, we find that it produces better results for this application than the more complicated random-effects model that accounts for study heterogeneity.

The general fixed effect model for meta-analysis is the inverse variance-weighted method [25]. Each study outcome measure is given a weight that is inversely proportional to its variance. For n independent studies, let T_i be the observed outcome measure with variance v_i and weight w_i. Then, an estimate of an aggregate outcome measure, given all studies, is calculated as follows:

$$\bar{T} = \frac{\sum_{i=1}^{k} w_i T_i}{\sum_{i=1}^{k} w_i} \qquad \text{where} \qquad w_i = \frac{1}{v_i}$$

In BN meta-analysis, we define the study outcome measure as the confidence estimates that are attached to each network edge. Thus, the fixed-effect meta-analysis model is applied to every network edge to obtain its combined confidence estimate. We treat the statistical confidence as an incidence rate (i.e. the proportion of networks in which a particular network edge exists). If the bootstrap approach is run m times resulting in m networks, then the statistical confidence, or incidence rate, for a particular edge e_{ij} that runs from node i to node j is $\frac{d_{ij}}{m}$ where d_{ij} is the number of networks where e_{ij} exists. Then, we define the outcome measure as the log incidence rate and its approximate variance [9] as:

$$log(T_{ij}) = log(d_{ij}/m), \qquad var(log(T_{ij})) = \frac{1}{d_{ij}}$$

This means that the meta-analysis weight is defined as:

$$w_{ij} = \frac{1}{v_{ij}} = d_{ij}$$

This type of meta-analysis is essentially a weighted averaging technique where edges are weighted using their own statistical confidence. Thus, edges with high confidences are strongly weighted and more likely to have a high confidence in the final Meta-analysis network.

Similarly to Consensus Bayesian networks, bootstrapping is used to generate the input individual networks that have confidences attached to each edge. In contrast to the consensus method, Bayesian network meta analysis does not require thresholding of the input networks to obtain PDAGs, since it directly combines the statistical confidences attached to each edge. However, the output meta-analysis network can be thresholded (using the same method that is described in section 2.1 for bootstrapped networks) to obtain a PDAG - and this is what we do to evaluate our meta-analysis networks.

3 Results

In this section we report on the experiments performed to evaluate the use of the Consensus and Meta-analysis approaches on multiple microarray datasets and compare them to the use of a single dataset and standard scale normalisation to combine the datasets. Initial experiments were carried out on a set of four datasets for a synthetic network of 13 genes. We then progressed to two real applications: E. coli and Yeast sub-networks.

For each application, every dataset was scale-normalised and a network with statistical confidences attached to each edge was learnt (using a bootstrapping approach with $m = 50$ iterations). A collection of aggregate networks were generated based on the individual dataset networks using the Meta-analysis and Consensus approaches. A single Meta-analysis network was constructed, where each edge has an attached statistical confidence. Multiple sets of Consensus networks were generated, each set corresponding to a different bootstrap statistical confidence threshold (0.1 to 0.9, at steps of 0.1) for the input networks generated from each dataset. This means that the input bootstrapped networks were all thresholded at the same value to form PDAGs, and these formed the input to the Consensus method. Each set of Consensus networks contains networks generated for each consensus threshold from 0 to 1, at steps of $1/n$ (where n is the number of datasets). Additionally, for comparison purposes, a bootstrapped network was learnt from a combined scale-normalised dataset (the Normalisation Only network).

We evaluate the learnt networks by comparing them to documented gene interactions. These were obtained from various sources according to the application. Whilst the synthetic network was fully known, E. coli regulatory interactions are documented in the online database RegulonDB [26] and yeast interactions (both confirmed and potential) are listed in the YEASTRACT database [27]. The learnt networks are compared to the true network in terms of true and false positives and negatives. A true positive (TP) is an edge that is present in both the learnt and true networks. A false positive (FP) is an edge that is present in the learnt network but not in the true network. A false negative (FN) is an edge that is in the true network but not in the learnt network, whilst a true negative (TN) is an edge that is not in the true or learnt network. In terms of the directionality of edges in the learnt network, if the direction conflicts with that in the true network, then the edge is counted as a FP. If the learnt network contains an undirected or unknown edge that is directed in the true network we count this as a TP. Whilst we do not want to 'miss' documented interactions (i.e. a low FN rate is desirable), a low FP rate is more important as FPs are significantly more costly to biologists. However the online databases from which our 'true' networks are extracted are limited to interactions that have been confirmed by biological studies. For example, RegulonDB contains regulatory information for only about 25% of the genes in the E. coli genome [28]. Therefore the proportion of FP interactions recorded in our learnt networks is likely to be higher than in reality.

In order to compare different approaches, it is common practice to use Receiver Operator Characteristic (ROC) curves. A ROC curve allows one to view graphically the performance of a classifier by plotting the TP rate (the proportion of true interactions that are identified) against the FP rate (proportion of incorrectly identified interactions):

$$TPrate = \frac{TP}{TP + FN} \qquad FPrate = \frac{FP}{FP + TN}$$

In a ROC space, a perfect network (i.e. identical to the true network) would have a TP rate of 1 and a FP rate of 0, which would sit at the top-left corner of the plot. For our experiments we plot a ROC curve where each point corresponds to a statistical confidence or a consensus threshold. For Meta-analysis each point of the ROC curve refers to the TP and FP rates of the PDAG extracted from the Meta-analysis network at different bootstrap confidence thresholds (from 0 to 1 at steps of 0.1). For Consensus, each point of the ROC curve refers to the TP and FP rates of the consensus network at different consensus thresholds (from 0 to 1, at steps of $1/n$). This means there are multiple ROC curves for the Consensus approach, each one constructed for a set of input networks obtained from a different bootstrap threshold. Since the Meta-analysis approach directly combines bootstrap confidences, and there is no initial thresholding step as for the Consensus approach, it has one ROC curve only.

A global measure of the classifier performance, often used in classification problems, is the Area Under the ROC Curve (AUC). AUC is a value between 0 and 1. We use the AUC to compare the networks generated by the various methods. In general, the closer the AUC is to 1 (and further away from 0.5) the better the overall performance of the network. The AUC measures *discrimination*, that is, the ability of the model to correctly classify instances (in this case, an instance is whether an interaction between a pair of genes exists). The AUC also specifies the probability that when we draw one positive and one negative example at random, a higher value is assigned to the positive than to the negative example. This direct interpretation of the AUC originates from the use of the ROC in applications where instances can be assigned a value or score that can be used to rank instances from most to least likely positive. For example, in medical studies where patients are classified into diseased and healthy and assigned a score based on the severity of their disease [29].

In order to obtain statistical estimates on the significance of the results, we ran this process several (15) times for each dataset. Thus, mean TP and FP rates (in order to estimate a mean ROC curve) and AUC measurements were obtained for each method. We use a paired t-test to compare the relative performances of the different approaches and measure whether the differences between their mean AUCs are statistically significant.

3.1 Synthetic network

The synthetic regulatory network consists of 13 genes as shown in Fig. 3. Four time-series expression datasets were generated for the network using differential equations to mimic a transcriptional network. The change of the expression of each gene is determined by a function composed of three parts: activation by a single other gene, repression by a single other gene and decay. Each of these parts has one parameter, which is (uniformly) randomly selected from a predefined range. Each dataset generated varies because the parameters for activation, inhibition and decay are chosen randomly for each gene, the predefined range of these parameters may vary, the perturbations vary and other parameters of the simulation (such as the length of the time lag) may also vary. Each dataset had a varying number of samples ranging from 200-600, as detailed in Table 1.

Figure 4 compares the difference in the mean AUC for each aggregation approach against each other and against the mean AUC of each individual dataset network (that are shown using horizontal lines). We also compare the combination of all datasets against the combination of a subset of the datasets (where the subset is chosen based on the performance of the networks). We refer to the networks generated by datasets 1-4 as Data1, Data2, Data3 and Data4 respectively, whilst the datasets themselves are referred to as dataset 1, dataset 2, dataset 3 and dataset 4.

Table 1: Summary of Synthetic datasets

Dataset	Number of Observations
1	200
2	400
3	600
4	600

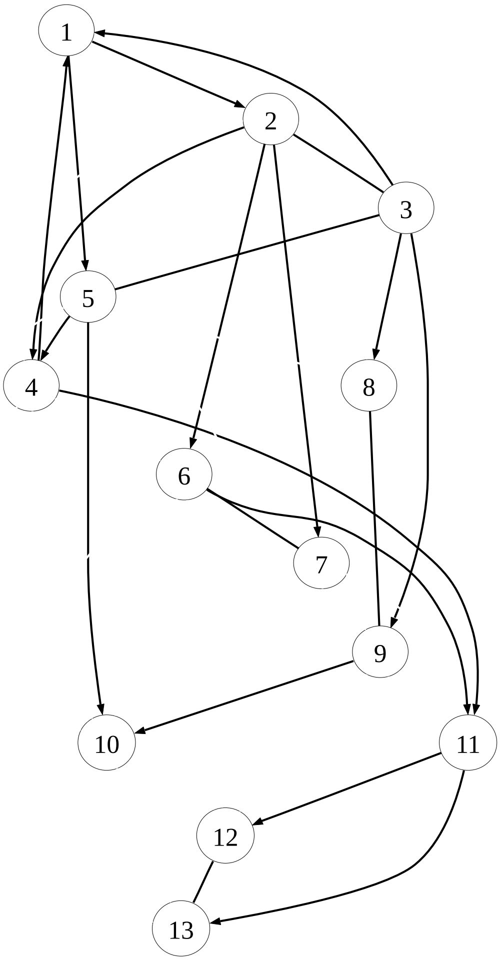

Figure 3: True synthetic network

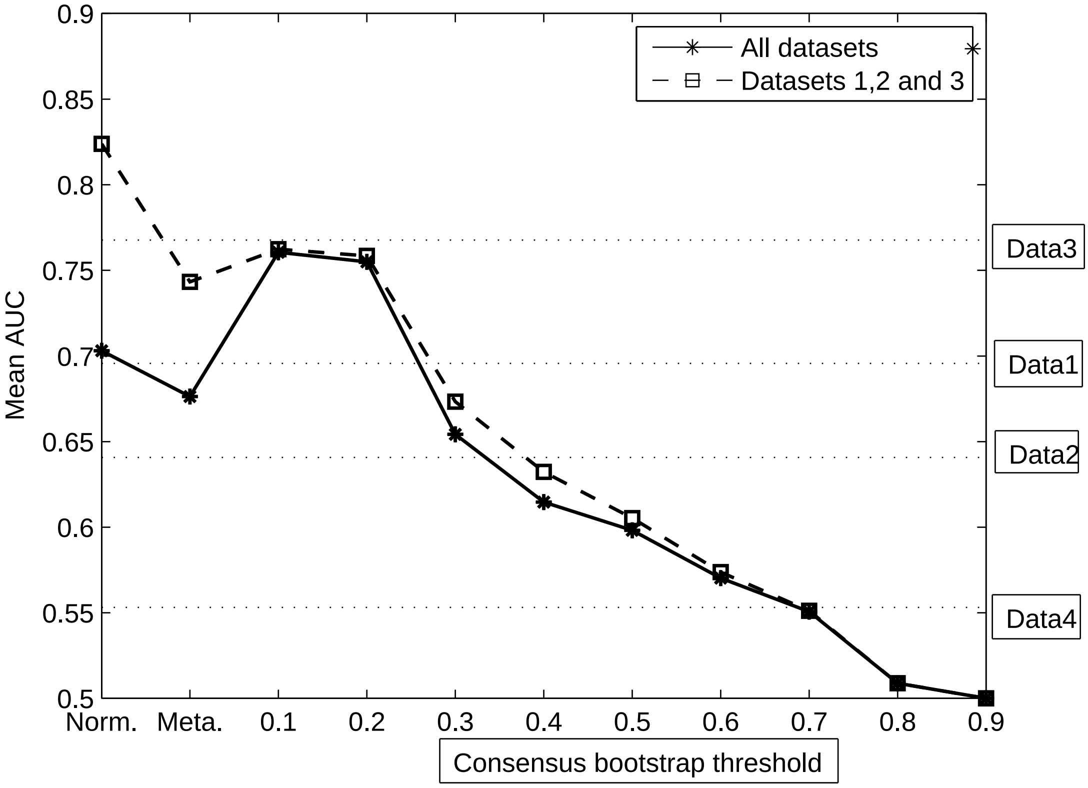

Figure 4: Mean AUC of learnt synthetic networks

Figure 4 shows that the Consensus approach performs best on the set of individual PDAGs extracted using a bootstrap threshold of 0.1. In this case the approach obtains a mean AUC of 0.76 (for a ROC curve that is obtained from set of Consensus networks, for Consensus thresholds from 0 to 1, at steps of 1/4 since there are $n = 4$ datasets). According to the paired t-test, this Consensus network set outperforms 3 of the 4 individual networks (Data1, Data2 and Data4), as well as the Normalisation Only and Meta-Analysis networks with statistical significance ($p < 0.01$). Meta-analysis, which obtains a mean AUC of 0.68, and Normalisation Only (obtaining a mean AUC of 0.70) only significantly outperform Data2 and Data4.

By selecting a consensus threshold we can obtain a single network structure from a set of Consensus networks. For example, a bootstrap threshold of 0.1 for the input networks, with a consensus threshold of 1.0 (where every edge in the Consensus network must appear in all input networks) provides the best TP and FP rates, which are 0.50 and 0.07 respectively (network not shown). In other words, it is able to identify half of the edges in the true network with a fairly low FP rate.

For the Consensus and Meta-analysis approaches, the robustness of an interaction can be identified using the confidence or consensus threshold attached to its edge. The ‘robustness’ of an edge in a Consensus network indicates in how many datasets it is found. Thus we can view a set of Consensus networks as a single network with each edge having a consensus threshold, or as a set of networks, each generated at a different Consensus threshold. The ‘robustness’ attached to a Meta-analysis edge is slightly different, as it incorporates the original bootstrapped confidences. In this case it represents the strength of the edge’s confidence over all the individual dataset networks. This is particularly useful for visualisation of the learnt networks. Fig. 5 shows the learnt Consensus network (obtained from input networks thresholded at a confidence threshold of 0.1) with edges shaded according to their consensus threshold. It can be seen by eye there is a correlation between the more robust edges and the true network (shown in Fig. 3).

Figure 4 shows that Data1, Data2 and Data3 are much closer to the true network than Data4, which we found contained many edges that were FPs. Upon closer inspection of dataset 4, we find that its randomly selected time lag length parameter is much larger than for the other datasets, perhaps explaining why Data4 performance is weaker. To eliminate the influence of dataset 4 we ran the Normalisation Only, Meta-analysis and Consensus approaches on datasets 1-3 only. Over the three datasets, Normalisation Only and Meta-analysis perform much better, their mean AUC increasing to 0.82 and 0.74 respectively. In fact, Normalisation Only outperforms all other networks with statistical significance $p < 0.01$, whereas the Consensus and Meta-analysis approaches are still unable to significantly outperform Data2. The difference between the performance of the Consensus and Meta-analysis approaches is no longer statistically significant. Since Data4 contains FP edges with high bootstrap confidences, Meta-analysis and Normalisation Only perform far more reliably when dataset 4 is removed from the input. By comparison, the Consensus approach is not so greatly affected by the removal of Data4 (see Fig. 4). Since the Consensus approach identifies consistencies across the set of individual dataset networks, it is able to discard the false positives introduced by Data4.

3.2 E. coli SOS response network

We consider an example of a single transcriptional module in E. coli - an SOS repair system. The module consists of approximately 30 genes and one transcriptional repressor, LexA. UV irradiation and other DNA damaging agents are known to trigger the induction of the stress-related SOS response, a coordinated increase in the level of expression in the set of genes, which is negatively regulated by LexA [30]. We selected a number of these genes (based on data availability) to form

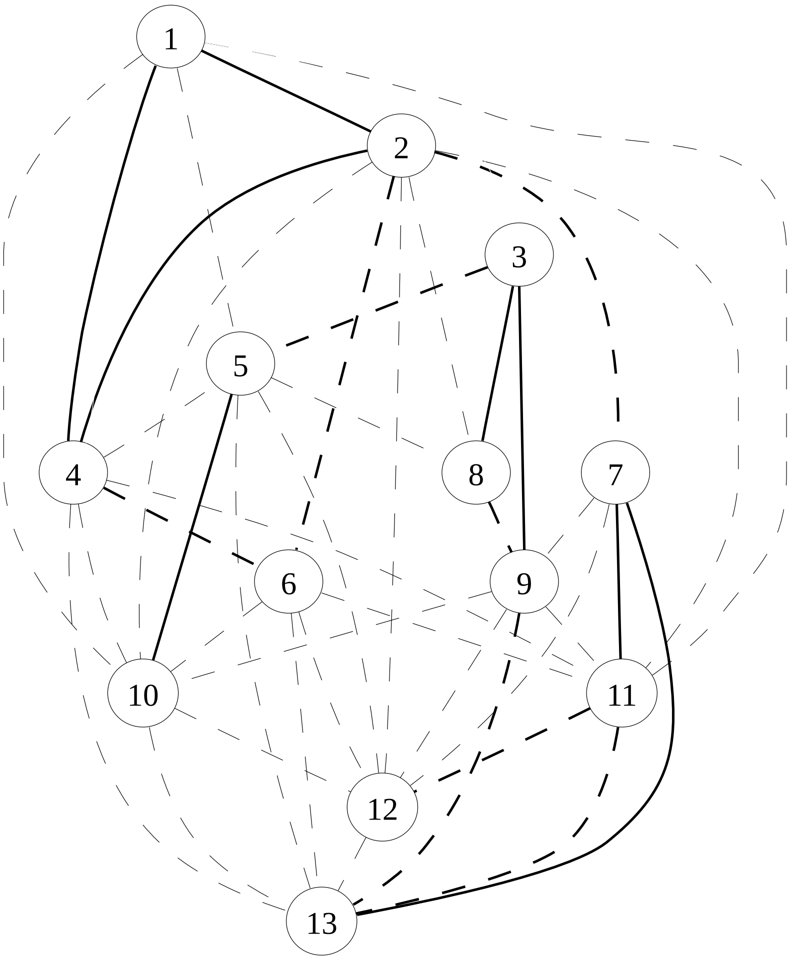

Figure 5: Synthetic consensus network, obtained from all input networks thresholded at a bootstrap confidence of 0.1. Edges are shaded or marked according to robustness - bold edges obtain a high consensus threshold (≥ 0.75). Bold and dashed edges have $0.50 \leq$ consensus < 0.75, whereas the dashed (only) edges have consensus ≤ 0.25.

Table 2: Summary of E. coli datasets

Dataset	Description	Platform	Number of Observations
Courcelle *et al.* [31]	UV irradiation	cDNA	15
Faith *et al.* [28]	Various	Affymetrix	254
Khil *et al.* [32]	DNA damage	cDNA	8
Sangurdekar *et al.* [33]	Various inc. UV irradiation	cDNA	240

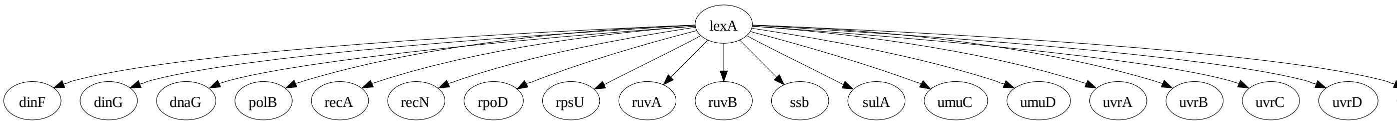

Figure 6: E. coli SOS response transcriptional module

a sub-network (see Fig. 6). Table 2 provides a summary of the four selected datasets, which are all focused on experiments related to SOS response. The datasets each originate from different research groups and microarray platforms including cDNA microarray technology and Affymetrix olignucleotide microarrays. For the Affymetrix data, in order to create an equivalent to cDNA microarray log ratio values, we subtracted the average log expression level of a gene from one experiment from the log expression level for that gene in a given experiment, allowing comparisons of different genes to each other.

Figure 7 compares the difference in the mean AUC for each aggregation approach against each other and against the mean AUC of each individual dataset network (that are shown using horizontal lines). We also compare the combination of all datasets against the combination of a subset of the datasets (where the subset is chosen based on the performance of the networks).

Figure 7 shows that the Consensus networks generated from sets of input networks thresholded at lower bootstrap confidences perform most successfully of the aggregation approaches (the best results are obtained with a bootstrap confidence threshold of 0.1). In this case the Consensus approach obtains a mean AUC of 0.58, outperforming three of the four individual dataset networks and the Normalisation Only and Meta-analysis approaches (with statistical significance $p < 0.01$). The low bootstrap threshold may be explained by the fact that there are very few edges with a high confidence (e.g. over 0.5 or 0.6) and these only occur in the Faith and Sangurdekar networks, for which the datasets contain a larger number of observations. Meta-analysis obtains a mean AUC of 0.52 (significantly outperforming only one of the four individual networks), whilst the mean AUC for Normalisation Only is just 0.47 and it is significantly outperformed by two of the individual dataset networks.

We believe that the nature of the SOS module plays a part in the high number of FP edges and relatively low AUC, in comparison to the results on synthetic data. It is a sparse network - in fact a Naïve Bayes model - and so all variables are correlated, becoming independent conditional on the regulator LexA. This makes it more difficult to identify spurious interactions. Figure 8 shows a Consensus network (with a consensus threshold of 1.0 and generated from input PDAGs calculated at bootstrap confidence threshold of 0.1). Whilst interactions between LexA and six genes are found, there are many other discovered interactions - e.g. the UVR family are obviously related. In previous experiments on the Courcelle dataset we were able to identify the regulator LexA consistently from a group of candidate transcription factors (regulator genes) for each target gene using BNs [34]. However, identifying the regulator when choosing from within a group of

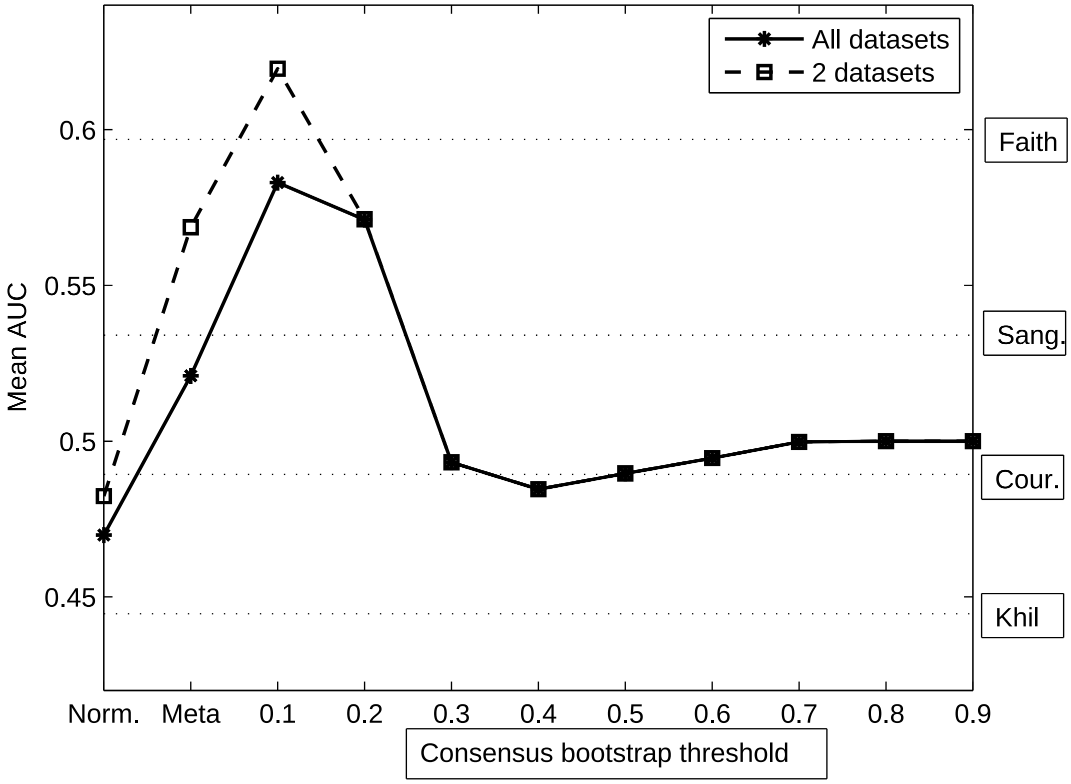

Figure 7: Mean AUC of learnt E. coli networks

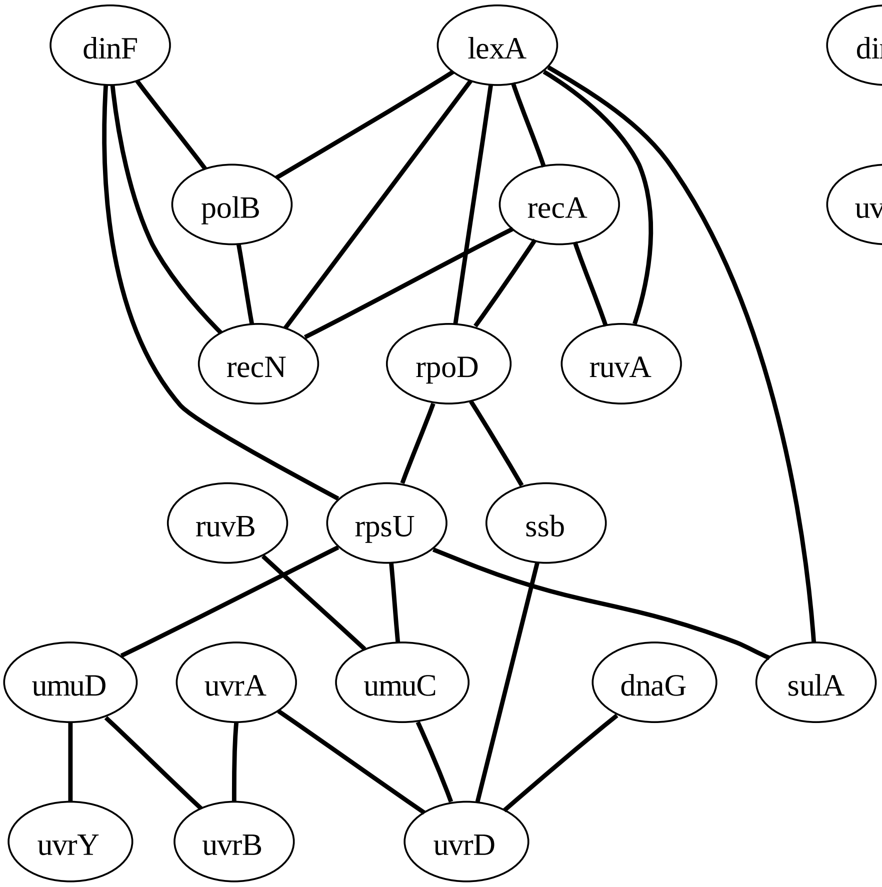

Figure 8: E. coli consensus network generated from the Faith and Sangurdekar datasets (each input network thresholded at a bootstrap confidence of 0.1) with a 1.0 consensus threshold (all edges appear in both input networks).

correlated genes is far more challenging. This of course also has a bearing on the calculation of FP edges between the learnt models and the 'true' network. In addition, it is likely that the 'true' network is in fact incomplete, which assists in explaining why the absolute performance of all networks is much lower in comparison to the synthetic data experiments.

Similarly to the synthetic data, some datasets perform better than others. In this case, the networks generated from datasets with relatively small numbers of observations - Courcelle and Khil - perform more weakly, their networks obtaining AUCs of 0.49 and 0.44 respectively. We ran Normalisation Only, Meta-analysis and Consensus on the Faith and Sangurdekar networks only. This improved the results for the Consensus approach, increasing the mean AUC to 0.62. It outperforms both the Faith and Sangurdekar networks with $p = 0.025$. Meta-analysis also makes an improvement, the mean AUC increasing from 0.52 to 0.57, but is unable to outperform the Faith network.

On synthetic data (especially on the three 'best' datasets), the simple Normalisation Only approach produced one of the best performing networks. However on the E. coli data, the Normalisation Only approach does not obtain such successful results. In fact, the Normalisation Only networks are the worst performing networks, and do worse in terms of AUC than 3 of the individual dataset networks. However, the synthetic data are not generated to contain any experimental or platform biases whereas these are inherent in the real E. coli data.

Table 3: Summary of yeast datasets

Dataset	Description	Platform	No. Obs
Beissbarth *et al.* [35]	Heat-shock response	cDNA	12
Eisen *et al.* [36]	Cold-shock and heat-shock response	cDNA	14
Gasch *et al.* [37]	Environmental changes inc heat-shock response	cDNA	173
Grigull *et al.* [38]	Heat-shock response	cDNA	27
Spellman *et al.* [39]	Cell-cycle	cDNA	73

3.3 Yeast heat stress network

We take the example of 9 transcription factors (TFs) related to heat-shock response from Wang *et al.* [12] in order to evaluate the algorithm on a sub-network of a manageable size and make a comparison between the two methods[2]. Two of the TFs selected (HSF1 and SKN7) are known to be directly involved in heat-shock response and are documented as regulating 4 TFs among the 9. The sub-network is shown in Fig. 9. We use microarray datasets that are publicly available on the YeastBASE expression database. Most selected are from studies that include heat-shock response experiments - see Table 3.

Figure 10 compares the difference in the mean AUC for each aggregation approach against each other and against the mean AUC of each individual dataset network (that are shown using horizontal lines). We also compare the combination of all datasets against the combination of a subset of the datasets (where the subset is chosen based on the performance of the networks).

Once again, the Consensus network set (generated from input networks at a low bootstrap confidence threshold of 0.1) obtain the best results of the aggregating approaches, outperforming all individual dataset networks, obtaining a mean AUC of 0.53. Using the paired t-test, we find this network set outperforms 3 of the 5 individual dataset networks with statistical significance $p < 0.01$. The Meta-analysis and Normalisation Only networks obtain mean AUCs of only 0.46 and 0.47 respectively. They are significantly outperformed by the Consensus network set and three of the five individual dataset networks.

Comparison of the AUC for each individual dataset network shows that three of the datasets perform noticeably poorly. If we remove these datasets from the input to the algorithms we find a marked improvement for all aggregation approaches (see Fig. 10). The Consensus approach obtains the best results, with a mean AUC of 0.55 whilst the individual networks for the Gasch and Spellman datasets obtain mean AUCs of 0.53 - a statistically significant difference with $p = 0.10$. In this case, we find the best Consensus networks are generated when the input PDAGs have been obtained by thresholding the boostrapped networks at relatively higher thresholds of 0.3 - 0.4. This is because the Gasch and Spellman networks have higher confidences attached to their edges than the networks generated from the other three datasets. The Meta-analysis and Normalisation Only approaches also show an improvement, so much so that there is no statistically significant difference in the AUC for the networks generated by them and the Consensus approach.

In Figure 10, we find that there is a significant dip in AUC at the 0.2 bootstrap threshold Consensus network. This is explained by that fact that there is a peak in edge confidences between 0.1 and 0.2 in the individual yeast networks (data not shown). Whilst a 0.2 thresholded individual PDAG includes the same edges as a 0.3 PDAG, a lower threshold means that more FP edges may

[2] In [12] they use a network of 10 TFs. We remove the gene SOK2 due to the many missing values in some of the datasets.

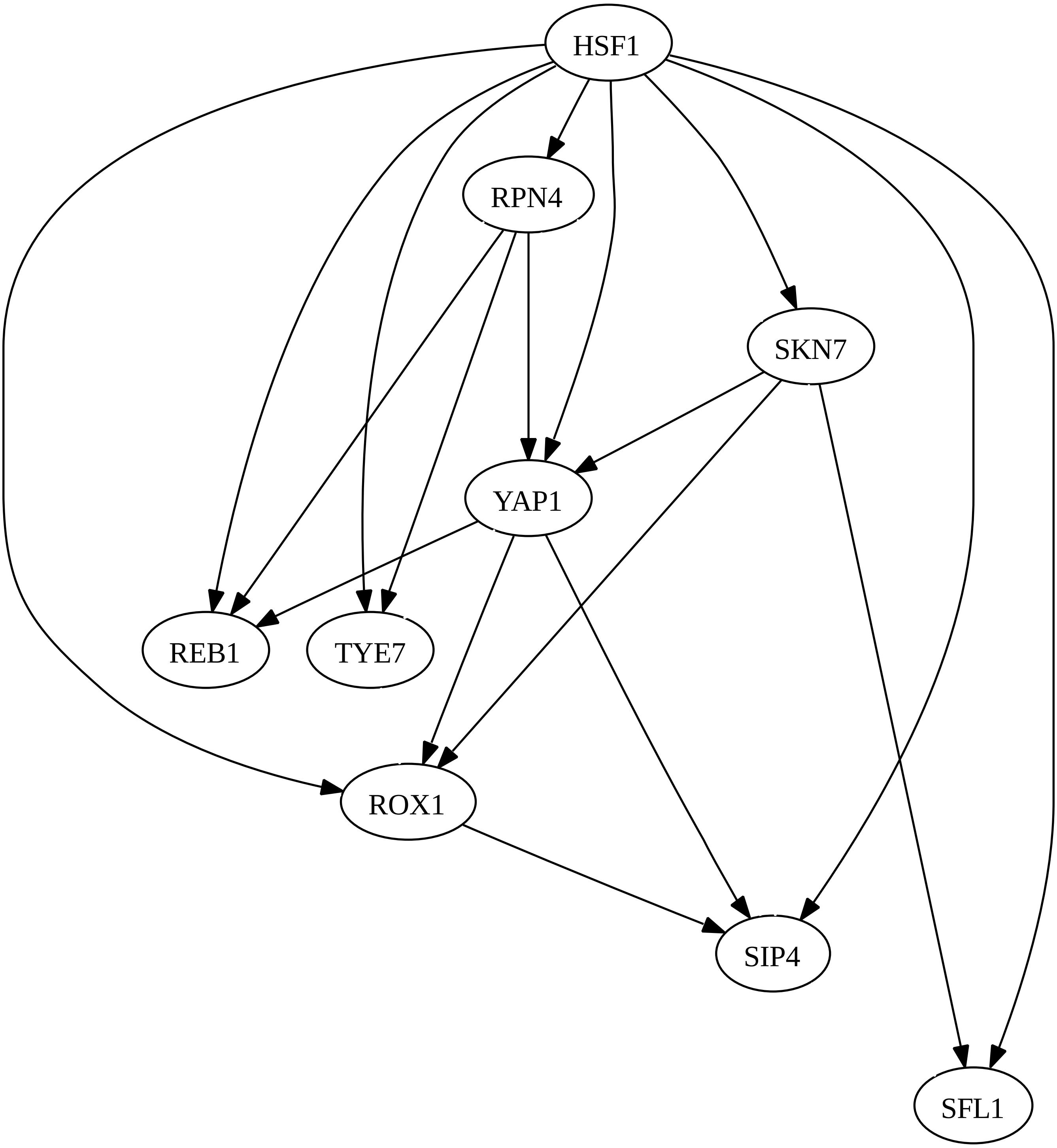

Figure 9: Yeast true network according to the YEASTRACT database (including confirmed and potential interactions).

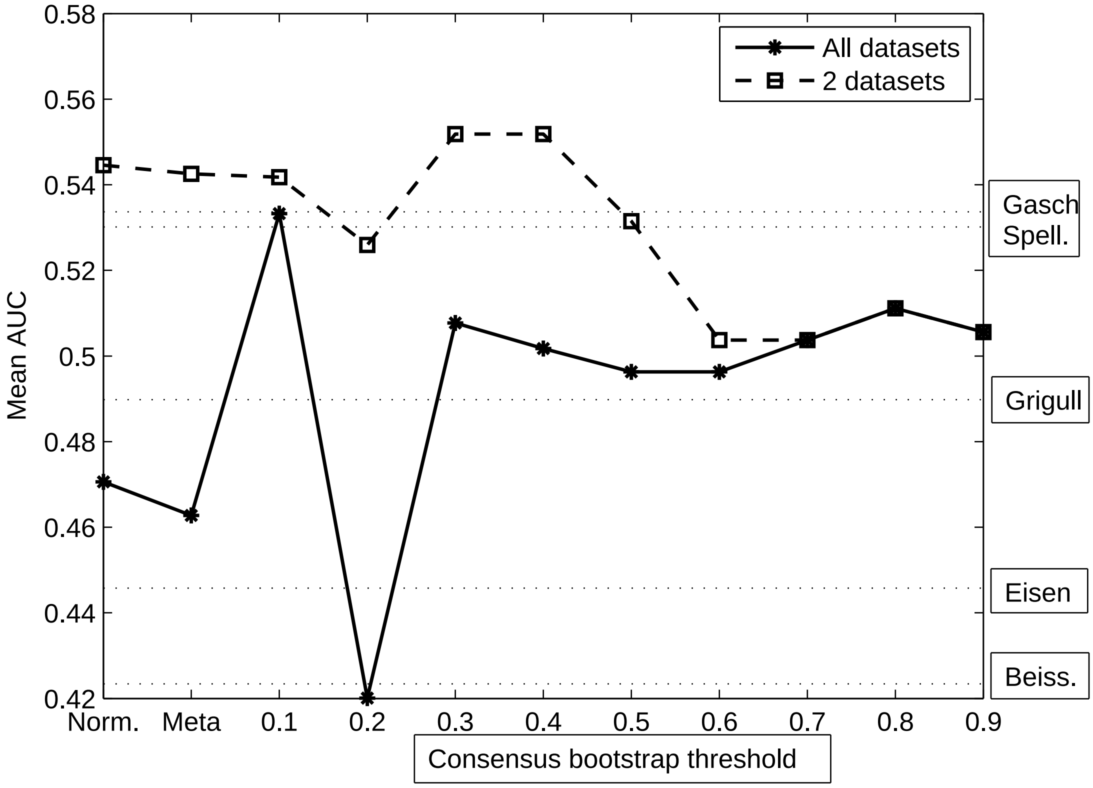

Figure 10: Mean AUC of learnt yeast networks

be included, causing the AUC to decrease. Similarly, lowering the threshold from 0.2 to 0.1, more edges are included, but in this case they are TP edges, causing an increase in AUC.

In comparison to the work by Wang *et al.* [12], both the Consensus and Meta-analysis networks are more successful based on our performance criteria. The Wang *et al.* network obtains a TP rate of 0.17 and a FP rate of 0.75. In comparison, our Consensus networks (from all datasets with a bootstrap threshold of 0.1) obtain a mean TP and FP rates of 0.58 and 0.54 respectively at a 0.8 consensus threshold and 0.16 and 0.09 at a 1.0 consensus threshold. Figure 11 shows such a Consensus network (0.8 threshold) that contains 13 TP edges and 7 FP edges. This network shows which edges are more robust (i.e. found in more individual dataset networks). We should also point out that Wang *et al.* only use some of the time-series in the Gasch dataset to generate their consensus network, whereas our consensus network is generated from a broader set of studies.

4 Conclusions

The purpose of this paper has been to investigate whether post-learning aggregation for generating GRNs from multiple microarray datasets (that is, learning models from each dataset and combining the models) can produce better results than concatenating the datasets after scale normalisation and then learning the model - a simple pre-learning aggregation method. We have presented two novel post-learning aggregation approaches for combining multiple microarray datasets to generate GRNs and compared them against scale normalisation.

Each of our new approaches is based on aggregating high-level features of BN models that have been generated from a set of individual microarray datasets. Thus, they possess the benefits of post-learning aggregation approaches, meaning they can be used to combine datasets generated by different platforms, research groups and laboratories and do not necessarily require normalisation of the datasets, which can be complicated on cross-platform microarray datasets. Meta-analysis BNs combine statistical confidences attached to network edges using an inverse-variance weighted method whilst Consensus BNs identify regulatory interactions that are found consistently across all datasets. Both methods produce networks with a measure of 'robustness' attached to each edge, which in a Consensus network indicates in how many datasets it is found. The 'robustness' attached to a Meta-analysis edge is slightly different, as it incorporates the original bootstrapped confidences. In this case it represents the strength of the edge's confidence over all the individual dataset networks.

We compared pre- and post-learning aggregation approaches with each other as well as against the performance of the individual dataset networks. On clean, unbiased synthetic data a simple Normalisation Only approach performs very well - significantly outperforming both Consensus and Meta-analysis networks and the individual dataset networks. However, on real data that is biased and generally noisier, this did not hold. In fact, Normalisation Only often performed worse than many of the networks generated from a single dataset. On E. coli data, we found that Meta-analysis and Consensus networks both provided a significant improvement over Normalisation Only. In particular, the Consensus approach increased the AUC by over 0.1. On the yeast sub-network, the absolute increase in AUC was not as great, but was still statistically significant. Thus, on the basis of the experiments presented in this paper, post-learning aggregation does provide an advantage over concatenating normalised datasets for learning from multiple *real* microarray datasets.

Whilst Consensus and Meta-analysis outperform Normalisation Only when learning from multiple microarray datasets, we also found that unless the worst-performing datasets were removed, the networks produced by post-learning aggregation approaches did not always outperform all the individual dataset networks. This leads to the question, is there a benefit to learning from multi-

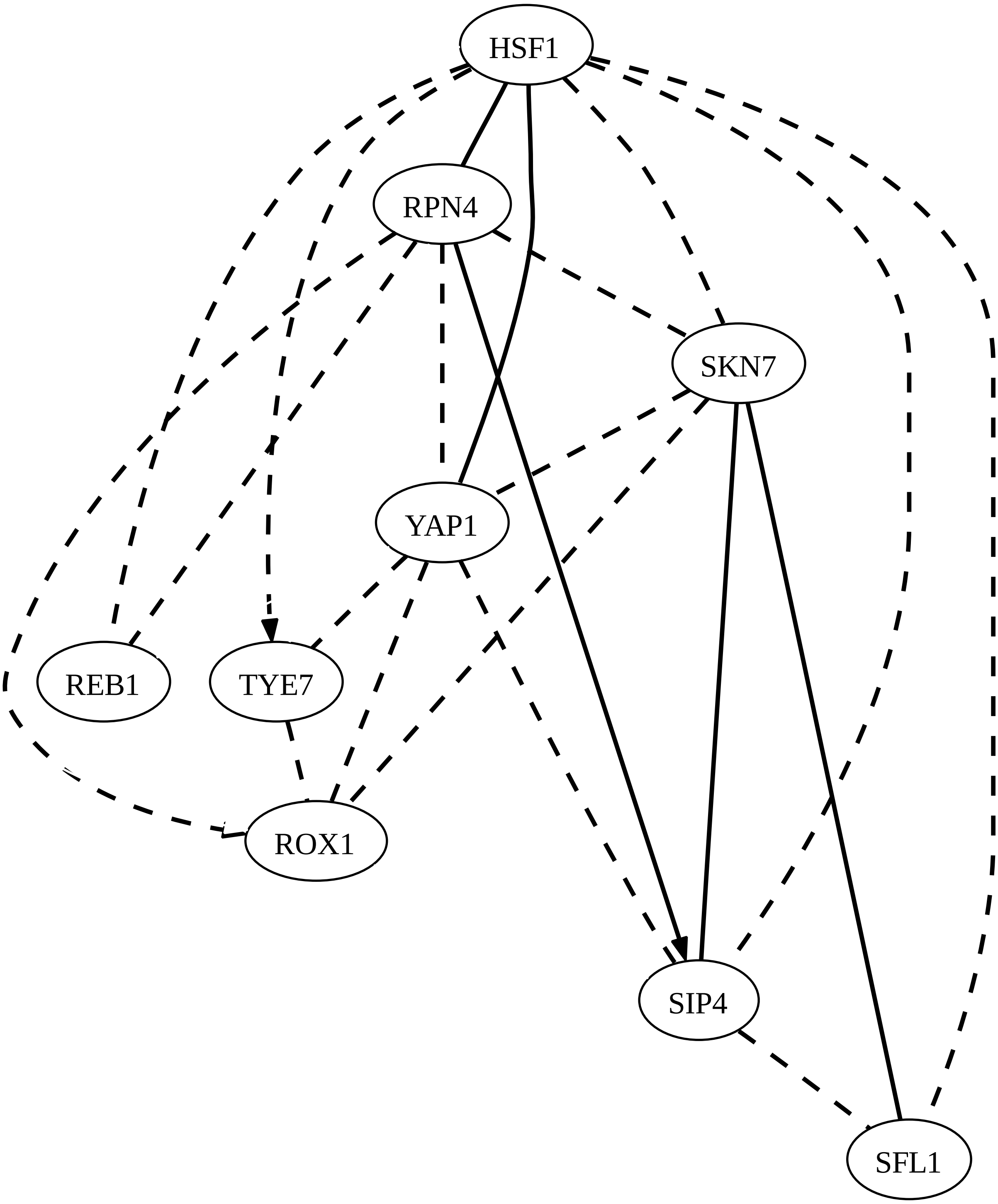

Figure 11: Consensus network (from all yeast datasets, each individual dataset network thresholded at a bootstrap confidence of 0.1). Dashed edges obtain 0.8 consensus whilst bold edges have a 1.0 consensus.

ple microarray datasets if the combined models do not outperform all individual dataset models? We believe so. When little is known about the datasets, post-aggregation learning can be used to identify the more robust and persistent interactions across datasets and filter out noisy and spurious relationships. The Consensus approach identifies consistencies amongst the collection of datasets and so it is least affected by poorly performing input networks. On the other hand, since Meta-analysis is a weighted-averaging technique, where edges with a high statistical confidence are given more influence, it can work well with only one well-performing dataset as the influence of lower confidence edges is weak. Conversely, however, its performance can be easily influenced by a single dataset that contains false positives and negatives with high statistical confidences.

We found that the datasets which generated the weakest performing networks were generally those with a small number of samples (at least, in the case of real data). Including these datasets with a small number of samples can actually have a negative effect by shifting focus from a larger dataset. Therefore it may be advantageous to only accept datasets with a larger number of samples, or at least to lessen the influence of datasets with a smaller number of samples.

It would also be desirable to reduce the number of parameters on the better-performing Consensus approach. When it is used in conjunction with bootstrapping to learn the input networks, the user is required to choose a bootstrap and a consensus threshold (although the final network can be viewed with edge 'robustness' as shown in the figures in this paper rather than choosing a consensus threshold). Meta-analysis is relatively simpler and 'parameter-free', since the bootstrap confidences are directly used to compute the aggregated network (however, if the user wishes to extract a PDAG, a threshold must be chosen).

Thus, there is room for improvement in the post-learning aggregation methods. A hybrid approach between Consensus and Meta-analysis is worth investigating. For example, the Meta-analysis approach could be modified to incorporate a Consensus term in the calculation of the combined outcome measures. Extra weighting could be applied to edges that have consistent confidences across all datasets, increasing their aggregated statistical confidence. This would assist in countering the problems of occasional high confidence FPs negatively influencing the final network in Meta-Analysis and the large number of parameters in the Consensus approach.

Additional further work will also involve extending the modelling techniques in a number of ways. Temporal information can be incorporated through the use of time nodes and dynamic BNs (this should improve the directionality of learnt interactions and allow cyclic behaviour to be introduced). Hidden nodes can be used to model unobserved variables. Furthermore, in these experiments we used datasets that were relevant to the network under consideration (for example, we used E. coli datasets from DNA damage experiments for the SOS response module). However we intend to investigate whether more diverse datasets could be combined by using additional nodes to represent the experiment type.

Since our approach is based on combining networks, it has the potential to integrate many heterogeneous types of data - provided that GRN models can be built from these datasets. We plan to look at the incorporation of other data sources or expert knowledge such as transcription factor binding sites, protein-protein interaction data and textual information extracted from scientific literature.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive criticisms that contributed to the final version of this paper. We would also like to acknowledge our colleague Michael Hirsch for generating the synthetic datasets.

References

[1] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.

[2] A.J. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Bayesian methods for elucidating genetic regulatory networks. *IEEE Intelligent Systems*, 17(2):37–43, 2002.

[3] D. Pe'er, A. Tanay, and A. Regev. MinReg: A scalable algorithm for learning parsimonious networks in yeast and mammals. *Journal of Machine Learning Research*, 7:167–189, 2006.

[4] L.A. Soinov. Supervised classification for gene network reconstruction. *Biochemical Society Transactions*, 31(6):1497–1502, 2003.

[5] W.P. Kuo, T.K. Jenssen, A.J.Butte, L.Ohno-Machado, and I.S. Kohane. Analysis of matched mRNA measurements from two different microarray technologies. *Bioinformatics*, 18(3):405–412, 2002.

[6] A.K. Jarvinen, S. Hautaniemi, H. Edgren, P. Auvinen, J. Saarela, O.P. Kallionoemi, and O. Monni. Are data from different gene expression microarrays comparable? *Genomics*, 83:1164–1168, 2004.

[7] C. Yauk, M.L. Nerndt, A. Williams, and G. Douglas. Comprehensive comparison of six microarray technologies. *Nucleic Acids Research*, 32(15), 2004.

[8] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kauffman, San Francisco, CA, USA, 1991.

[9] A. Sutton, K. Abrams, D. Jones, T. Sheldon, and F. Song. *Methods for Meta-Analysis in Medical Research.* Wiley, Chichester, UK, 2000.

[10] S.K. Ng, S.H. Tan, and V.S. Sundararajan. On combining multiple microarray studies for improved functional classification by whole-dataset feature selection. *Genome Informatics*, 14:44–53, 2003.

[11] E.M. Conlon, J.J. Song, and J.S. Liu. Bayesian models for pooling microarray studies with multiple sources of replications. *BMC Bioinformatics*, 7(247), 2006.

[12] Y. Wang, T. Joshi, X.S. Zhang, D. Xu, and L. Chen. Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics*, 22(19):2413–2420, 2006.

[13] P. Stoica. On information criteria and the generalized likelihood ratio test of model order selection. *IEEE Signal Processing Letters*, 11(10), 2004.

[14] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[15] J. Pearl and T. Verma. A theory of inferred causation. In *Proceedings of Knowledge Representation and Reasoning 2*, pages 441–452, New York, USA, 1991. Morgan Kauffman.

[16] D. Chickering. A transformational characterization of equivalent Bayesian network structures. In *Proceedings of Uncertainty in Artificial Intelligence 11*, 1995.

[17] N. Friedman, M. Goldszmidt, and A. Wyner. Data analysis with Bayesian networks: A bootstrap approach. In *Proceedings of 15th Annual Conference on Uncertainty in Artificial Intelligence*, 1999.

[18] B. Efron and R. Tibshirani. *An introduction to the Bootstrap.* 1993.

[19] G. Smyth and T. Speed. Normalization of cDNA microarray data. *Methods*, 31:265–273, 2003.

[20] D. Park and X. Wang. Toward a general framework for microarray data comparison. In *Proceedings of the 6th IEEE International Conference on Computer and Information Technology (CIT'06)*, 2006.

[21] D.M. Pennock and D. Wellman. Graphical representations of consensus belief. In *Proceedings of Uncertainty in Artifical Intelligence 15*, pages 531–538. Morgan Kauffman, 1999.

[22] I. Matzkevich and B. Abramson. The topological fusion of Bayes nets. In *Proceedings of Uncertainty in Artificial Intelligence 8*, pages 191–198. Morgan Kauffman, 1992.

[23] D. Pe'er, A. Regev, G. Elidan and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 1(1):1-9, 2001.

[24] E. Peeling and A. Tucker. Consensus gene regulatory networks: combining multiple microarray gene expression datasets. In *AIP Conference Proceedings vol. 940, The 3rd International Symposium on Computational Life Sciences (COMPLIFE 2007)*, 2007.

[25] R. DerSimonian and N. Laird. Meta-analysis in clinical trials. *Controlled Clinical Trails*, 7:177–188, 1986.

[26] H. Salgado, S. Gama-Castro, M. Peralta-Gil, E. Diaz-Peredo, F. Sanchez-Solano, A. Santos-Zavaleta, I. Martinez-Flores, V. Jimenez-Jacinto, C. Bonavides-Martinez, J. Segura-Salazar, A. Martinez-Antonio, and J. Collado-Vides. Regulondb (version 5.0): Escherichia coli k-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Research*, 34, 2006.

[27] M. Teixeira, P. Monteiro, P. Jain, S. Tenreiro, A.R. Fernandes, N.P. Mira, M. Alenquer, A.T. Freitas, A.L. Oliveira, and I. S-Correia. The YEASTRACT database: a tool for the analysis of transcription regulatory associations in Saccharomyces cerevisiae. *Nucleic Acids Research*, 34:D446–D451, 2006.

[28] J.J. Faith, B. Hayete, J.T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J.J. Collins, and T.S. Gardner. Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biology*, 5(1), 2007.

[29] J.A. Hanley and B.J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.

[30] P. Quillardet, M.A. Rouffaud, and P. Bouige. DNA array analysis of gene expression in response to UV irradiation in Escherichia coli. *Research in Microbiology*, 154:559–572, 2003.

[31] J. Courcelle, A. Khodursky, B. Peter, P.O. Brown, and P.C. Hanawalt. Comparative gene expression profiles following UV exposure in wild-type and SOS-deficient Escherichia coli. *Genetics*, 158(1):41–64, 2001.

[32] P. Khil and R. Camerini-Otero. Over 1000 genes are involved in the DNA damage response of Escherichia coli. *Molecular Microbiology*, 44(1):89–105, 2002.

[33] D. Sangurdekar, F. Srienc, and A.B. Khodursky. A classification based framework for quantitative description of large-scale microarray data. *Genome Biology*, 7, 2006.

[34] E. Peeling, A. Tucker, and P.A.C t'Hoen. Discovery of local regulatory structure from microarray gene expression data using Bayesian networks. In *Proceedings of the annual workshop on Intelligent Data Analysis in bioMedicine And Pharmacology (IDAMAP)*, 2007.

[35] T Beissbarth, K Fellenberg, B Brors, R Arribas-Prat, J Boer, NC Hauser, M Scheideler, JD Hoheisel, G Schutz, A Poustka, and M Vingron. Processing and quality control of dna array hybridization data. *Bioinformatics*, 16(11), 2000.

[36] MB Eisen, PT Spellman, PO Brown, and D Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25):14863–8, 1998.

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

[38] J Grigull, S Mnaimneh, J Pootoolal, MD Robinson, and TR Hughes. Genome-wide analysis of mrna stability using transcription inhibitors and microarrays reveals post-transcriptional control of ribosome biogenesis factors. *Mol. Cell*, 24(12):5534–47, 2004.

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