

## Gene expression

# Predicting gene targets of perturbations via network-based filtering of mRNA expression compendia

Elissa J. Cosgrove<sup>1,†</sup>, Yingchun Zhou<sup>2,†,‡</sup>, Timothy S. Gardner<sup>1</sup> and Eric D. Kolaczyk<sup>2,\*</sup><sup>1</sup>Department of Biomedical Engineering and <sup>2</sup>Department of Mathematics and Statistics, Boston University, Boston, MA 02215, USA

Received on December 23, 2007; revised on August 11, 2008; accepted on September 4, 2008

Advance Access publication September 8, 2008

Associate Editor: Joaquin Dopazo

**ABSTRACT**

**Motivation:** DNA microarrays are routinely applied to study diseased or drug-treated cell populations. A critical challenge is distinguishing the genes directly affected by these perturbations from the hundreds of genes that are indirectly affected. Here, we developed a sparse simultaneous equation model (SSEM) of mRNA expression data and applied Lasso regression to estimate the model parameters, thus constructing a network model of gene interaction effects. This inferred network model was then used to filter data from a given experimental condition of interest and predict the genes directly targeted by that perturbation.

**Results:** Our proposed SSEM–Lasso method demonstrated substantial improvement in sensitivity compared with other tested methods for predicting the targets of perturbations in both simulated datasets and microarray compendia. In simulated data, for two different network types, and over a wide range of signal-to-noise ratios, our algorithm demonstrated a 167% increase in sensitivity on average for the top 100 ranked genes, compared with the next best method. Our method also performed well in identifying targets of genetic perturbations in microarray compendia, with up to a 24% improvement in sensitivity on average for the top 100 ranked genes. The overall performance of our network-filtering method shows promise for identifying the direct targets of genetic dysregulation in cancer and disease from expression profiles.

**Availability:** Microarray data are available at the Many Microbe Microarrays Database (M3D, <http://m3d.bu.edu>). Algorithm scripts are available at the Gardner Lab website (<http://gardnerlab.bu.edu/SSEMLasso>).

**Contact:** kolaczyk@math.bu.edu

**Supplementary information:** Supplementary Data are available at *Bioinformatics* on line.

## 1 INTRODUCTION

DNA microarrays have been applied to investigate genome-wide expression changes in response to a drug or to compare expression in a diseased cell population to that of normal cells. In many cases,

the goal of these studies is to identify direct gene targets of the perturbations, e.g. genes whose products are the molecular targets of a drug compound, or genes that are mutated or dysregulated in cancer or disease. However, hundreds to thousands of additional genes exhibit secondary responses due to changes in the activity of the relatively few direct targets (Courcelle *et al.*, 2001; Hughes *et al.*, 2000; Miklos and Maleszka, 2004). Many methods have been applied to address these ambiguities inherent in expression profiles (e.g. Golub *et al.*, 1999; Hughes *et al.*, 2000; Marton *et al.*, 1998; Natsoulis *et al.*, 2005; Subramanian *et al.*, 2005). Yet, these approaches do not explicitly account for interactions between genes, a major source of these indirect effects.

In recent work, di Bernardo *et al.*, 2005 proposed mode-of-action by network identification (MNI), in which a model of the network of interactions between genes was inferred using a large microarray compendium. This model was subsequently applied to ‘filter’ expression profiles of the perturbations of interest in order to identify the direct gene target(s) of each. MNI performed well in identifying the gene targets of promoter insertions and drugs, and was also applied successfully for identification of a known genetic mediator of prostate cancer (Ergün *et al.*, 2007). However, the method lacks a formal statistical framework and optimization criteria, resulting in poor generalization and requires a non-trivial amount of expert supervision to tune it appropriately.

In this article, we propose the formal statistical modeling framework of sparse simultaneous equation models (SSEMs), and an associated inferential strategy, for the problem of predicting directly perturbed genes from DNA microarray expression profiles. Simultaneous equation models (SEMs) consist of a system of equations where certain variables act as both dependent and independent variables. As such, SEMs are natural here for modeling the effects of gene–gene interactions because each gene is potentially influenced by every other gene. Furthermore, due to both the low numbers of regulators per gene (Jeong *et al.*, 2000, 2001) and a similarly low expected number of directly targeted genes for a typical perturbation, the model of network interaction effects is expected to be sparse—thus our emphasis on SSEMs.

Within our framework, the notion of a ‘directly targeted gene’ is interpreted mathematically to refer to a gene experiencing upon perturbation an additive shift in its mean level of mRNA expression, adjusted for the regulatory effects on that gene of all other genes. Furthermore, we implicitly have in mind single, isolated gene targets, such as when a gene is mutated or dysregulated in cancer or

\*To whom correspondence should be addressed.

<sup>†</sup>The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

<sup>‡</sup>Present address: National Institute of Statistical Sciences (NISS), Durham, NC 27709 USA.

disease, or small subsets of ‘localized’ gene targets, such as when perturbation by a drug compound affects a single molecular target and/or certain closely related genes downstream (e.g. via a signaling pathway). This particular sense of a ‘direct gene target’ and its response will of course match the mechanism of action of some types of perturbations better than others. We return to this important issue periodically throughout the article.

Our approach to the prediction of direct gene targets is two-stage in nature, involving the inference of the network interaction effects in our SSEM, followed by a residual analysis of perturbation data filtered to exclude these inferred effects. To infer the interaction effects, we applied Lasso regression (Tibshirani, 1996), which is particularly useful for sparse regression with large numbers of covariates, and has previously been applied for inferring the topology of gene regulatory networks (Bonneau *et al.*, 2006).

We applied our proposed method, SSEM-Lasso, to simulated data and three microarray compendia. We compared the performance of our algorithm to a ‘null’ compendium  $z$ -test method and to the MNI algorithm. Our algorithm demonstrated consistent improvement in predicting targets in simulated datasets, with a 167% increase in sensitivity on average for the top 100 ranked genes, compared with the next best method. The SSEM-Lasso algorithm also demonstrated a 15% improvement in sensitivity on average, for the top 100 ranked genes, in identifying the targets of genetic perturbations (deletion, mutation, overexpression) in two Affymetrix microarray compendia. The results presented herein clearly demonstrate promise for determining the genetic basis of disease based on expression profiles.

## 2 ALGORITHM

### 2.1 An SSEM for RNA concentrations

Motivated by a differential equations perspective, and using a simplification of Hill-type transcription kinetics (as described in Liao *et al.*, 2003), we modeled the concentration of each mRNA transcript measured in a single microarray experiment in the absence of perturbation as a weighted sum of the concentrations of all other transcripts

$$y_i = \sum_{j \neq i}^p \beta_{ij} y_j + \epsilon_i \quad (1)$$

where  $y_i$  is the concentration of transcript  $i$ ,  $\beta_{ij}$  represents the influence of transcript  $j$  on transcript  $i$ ,  $p$  is the number of transcripts measured and  $\epsilon_i$  is a stochastic term associated with the observation  $y_i$ . A detailed description of the derivation of our model is provided in the Supplementary Material.

The objective of our algorithm is to predict the direct targets of a perturbation of interest, e.g. genes that have been deleted or over-expressed, genes whose products are the molecular targets of a drug compound or genes that are mutated or dysregulated in cancer or disease. We assume that these direct influences of the perturbation can be modeled as an additive term in our expression for mRNA concentrations in a single microarray experiment i.e.

$$y_i = \sum_{j=1}^p \beta_{ij} y_j + \phi_i + \epsilon_i \quad (2)$$

where  $\phi_i$  is the direct influence of the perturbation on gene  $i$ .

When all genes are considered simultaneously over all  $n$  observed experimental conditions, the model can be written succinctly as

$$Y = BY + \Phi + \mathcal{E}, \quad (3)$$

where  $Y$  is the  $p \times n$  experimental data matrix,  $B$  represents the influence of each gene on every other gene (all  $\beta_{ij}$  terms),  $\Phi$  is a  $p \times n$  matrix of the values  $\phi_i$  and  $\mathcal{E}$  is a  $p \times n$  matrix of the corresponding stochastic terms  $\epsilon_i$ . Equation (3) is a SEM, as each measured expression level  $y_i$  is potentially influenced by each of the others.

We expect most genes to be regulated by only a small subset of the total genes measured, and thus assume that the matrix  $B$  describing these gene–gene interactions will be sparse. Similarly, we expect only a small number of genes to be directly targeted by a perturbation, and assume that the matrix  $\Phi$  is also sparse. Thus, our model is a sparse SEM (SSEM).

### 2.2 Prediction of the direct targets of a perturbation

Our goal is to identify significant entries of  $\Phi$  for perturbations (experimental conditions) of interest. If we knew the matrix  $B$  of network interaction effects, we could form the residuals for a given condition ‘*pert*’ based on the corresponding observations  $y^{pert}$

$$\mathbf{r} = y^{pert} - By^{pert} = \phi^{pert} + \epsilon^{pert} \quad (4)$$

and the task of detecting the direct gene target(s) could be viewed as one of detecting a sparse signal in a background ‘noise’. That is, we could filter out the network effects and perform a residual analysis. However, we do not know  $B$ , and thus it must also be inferred from the data.

Our approach to inference of  $B$  was selected to exploit the sparseness expected in both  $B$  and  $\Phi$ . We consider our data to follow a model (approximately) of the form  $Y \simeq BY + \mathcal{E}$ , temporarily ignoring the effects of the highly sparse matrix  $\Phi$ . A natural approach for solving this sparse regression problem is the Lasso (Tibshirani, 1996), which looks for the estimate that minimizes a combination of model lack-of-fit and the sum of the absolute values of the coefficients. Employing this method, we infer  $B$  row by row as

$$\hat{B}_{i,\cdot} = \arg \min_{B_{i,\cdot}: B_{i,i}=0} \sum_{j=1}^n (Y_{ij} - B_{i,\cdot} Y_{\cdot j})^2 + \lambda_i \|B_{i,\cdot}\|_1 \quad (5)$$

for  $i = 1, \dots, p$  where  $\|B_{i,\cdot}\|_1 = \sum_{k=1}^p |B_{ik}|$ . The parameter  $\lambda_i$  is a regularization or smoothing parameter that is determined for each row  $i$ . We use the LARS algorithm (Efron *et al.*, 2004) to solve the Lasso optimization in (5). We apply standard 10-fold cross-validation (e.g. as described in Hastie *et al.*, 2001, Ch. 7.10) to select the parameters  $\lambda_i$ .

Using our estimate of  $B$  (our network filter), we compute the residuals for conditions of interest

$$\hat{\mathbf{r}}^{pert} = y^{pert} - \hat{B}y^{pert} \quad (6)$$

following (4). We conduct outlier analysis by ranking the values in  $\hat{\mathbf{r}}^{pert}$  by their absolute values. Genes with higher ranks are considered more likely to be the direct gene targets of the perturbation. More sophisticated methods, such as those based on false discovery rates (FDR), could be used as well to annotate the rankings with a measure of importance and to declare targets accordingly. However, for our proposed method, we did not find

such results to differ noticeably from those based on ranks for the type of performance evaluation conducted within this study (see Supplementary Material, Figure S4), and the use of ranks here facilitates our comparison with other methods below. Nevertheless, we note that the values generated by these more sophisticated methods (e.g.  $Q$ -values based on FDR) may be useful for providing a more refined relative comparison among top candidate genes on a list for a given perturbation of interest, such as when seeking an objective cut off, as they provide relative magnitudes for each gene in the list while rank alone does not convey this information.

### 3 METHODS

#### 3.1 Simulated data

Simulated datasets were constructed in the following steps: (i) generate an interaction matrix  $B$ , (ii) generate unperturbed (or perturbed) ‘training’ data from this network and (iii) generate perturbed ‘test’ data with single-gene perturbations.

We generated the matrix  $B$  according to either a random or scale-free network of dimension 1000 genes by 1000 genes. We used the Matlab function `sprand` to generate sparse Erdős–Rényi random graphs at a specified density of non-zero entries (sparse rate). Barabási–Albert scale-free graphs were generated following Ravasz *et al.*, 2002, and the sparse rate was set by specifying a maximum in-degree for each gene. In both cases,  $B$  was assigned non-zero entries for pairs of nodes adjacent in the network (edges), with the entries drawn uniformly between  $-1$  and  $1$ . The diagonal was set to zero in accordance with our model. Networks were simulated to have approximate densities of 0.01 and 0.005.

We simulated the observations  $Y$  according to (3) assuming normally distributed input:

$$Y = (I - B)^{-1}(\Phi + \mathcal{E}) \quad (7)$$

where  $I$  is the  $p \times p$  identity matrix.

Separate data were generated for estimating the matrix  $B$  (‘training’) and evaluating performance (‘test’). ‘Unperturbed’ (noise only,  $\Phi$  set to zero) training data of dimension 1000 genes by 100 observations were generated by setting the matrix  $\Phi$  to zero so that only noise was propagated through the network. We considered this unperturbed dataset to be closest to ideal for estimating the  $B$  matrix because only network influences affect these data. Test data of dimension 1000 genes by 1000 observations were generated such that each column of  $\Phi$  had only one non-zero entry of value  $\phi$  and each gene was perturbed once (‘perturbed’ data). This simulates the situation when each perturbation has only one target, with influence  $\phi$ . These could be considered gene knockdown or overexpression perturbations, depending on the sign of  $\phi$  (we found that the sign of  $\phi$  did not affect our results). We took the ratio  $\phi/\sigma$  as a rough measure of the signal-to-noise ratio (SNR) in the data, where  $\sigma$  is the standard deviation of the terms  $\epsilon_i$ , and set the ratio to be 4, 16 and 100. For each set of simulation parameters (SNR, density, network type), we generated 10 sets of training and test data as described above, and plots presented in the Results section include performance over all 10 datasets.

In later simulations, we generated training data that consisted of single-gene perturbations rather than noise only, again of dimension 1000 genes by 100 observations (generated similarly to test data), which we expected to more closely represent the real data than the unperturbed training data described above, because both network interactions and direct perturbation influences will be present in these data. In these cases, the test data included both the training data and additional test data (Fig. 2).

We also generated test data in which 10 or 50 genes were perturbed to simulate a perturbation with multiple targets. Target genes were either selected at random (‘random’) or using the true interaction matrix  $B$  to select all genes directly downstream of a first gene chosen at random, and all genes directly downstream of these genes, and so on, until the desired number of targets was reached (‘umbrella’). A perturbed target was assigned a value  $\phi$  in its corresponding vector  $\Phi$ . For each number of multiple targets (10 and 50)

and target selection method (random and umbrella), 100 perturbations were simulated. Performance using these test data was evaluated using estimates of  $B$  derived from unperturbed training data.

#### 3.2 Experimental data

We used three microarray compendia to test our proposed algorithm. For each compendium, profiles with known perturbed targets were selected to form test sets for evaluation of performance; in most cases, experiments used for evaluating algorithm performance were also used in the network training phase.

The two-color yeast *Saccharomyces cerevisiae* compendium included 515 two-color cDNA microarray experiments: 300 from Hughes *et al.*, 2000 and 215 from Mnaimneh *et al.*, 2004, each representing a distinct experimental condition. All 515 experiments were used in the training phase. Performance was tested on 501 genetic perturbation experiments included in the compendium (275 deletions and 226 promoter insertions) and 11 drug perturbation experiments with known targets (eight from this compendium and the three additional drug experiments used in di Bernardo *et al.*, 2005).

The Affymetrix yeast *S. cerevisiae* compendium included 962 Affymetrix Yeast Genome S98 high-density oligonucleotide arrays, representing 465 experimental conditions (Table 1). The raw .CEL data files, taken from 66 externally conducted projects, were collected from National Center for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO) (Barrett *et al.*, 2007), European Bioinformatics Institute (EBI) ArrayExpress (Parkinson *et al.*, 2007) or directly requested from investigators. A description of the projects included in this compendium, as well as their references, are presented in Supplementary Material Table S1. Robust multichip average (RMA) (Irizarry *et al.*, 2003) was applied to normalize raw probe intensities to probe-set expression levels. Only probe-sets for non-dubious SGD-annotated open reading frames (ORFs) were included in analysis (6681 of 9335 total probe-sets). Additionally, in the interest of cutting down on computational demands, only initial and final timepoints from time-series experiments were used in the network training phase, yielding a final data matrix of 6681 probe-sets by 647 arrays (representing 285 experimental conditions) used for the network training phase. Testing on an earlier version of this compendium demonstrated that there was very little difference in performance when a subset of 465 non-time-series arrays were used instead of all 701 arrays (data not shown). This finding was consistent with work presented in Faith *et al.*, 2007a, where they found

**Table 1.** Composition of the Affymetrix compendia and the test sets constructed for evaluating performance

	Affymetrix compendia	
	Yeast	<i>E. coli</i>
Data comprising compendia		
Microarray experiments	962 (647) <sup>a</sup>	524
Experimental conditions	465 (285) <sup>a</sup>	264
Genetic perturbation test set		
Unique experimental conditions	169	130
Unique gene targets	61	66
Total targets tested	235	136
Drug perturbation test set		
Unique experimental conditions	41	105
Unique drug treatments	11	5
Unique gene targets	18	34
Total targets tested	141	502

Note that some experiments included multiple perturbations, and some genes were represented by multiple probe-sets.

<sup>a</sup>The number in parentheses indicates the subset of the data used for the network training phase.

that only a subset of arrays were required to reconstruct a network with nearly equivalent precision. The timepoint experiments excluded during the training phase were included in the testing phase. Details of the genetic and drug perturbation test sets constructed from this compendium are presented in Table 1. For conditions sampled in replicate, the median Lasso residual values were used in ranking targets. This compendium is available for browsing and download as *yg\_s98\_v3\_Build\_2* at the Many Microbe Microarrays Database (M3D) (<http://m3d.bu.edu>) (Faith *et al.*, 2007b).

The Affymetrix *Escherichia coli* compendium included 524 Affymetrix *E. coli* Antisense Genome v2 high-density oligonucleotide arrays, representing 264 experimental conditions. The majority of this compendium was assembled from 10 publications in Faith *et al.*, 2007a, and the additional arrays were from Dwyer *et al.*, 2007 and Kohanski *et al.*, 2007. Raw probe intensities were RMA-normalized to probe-set expression levels using a re-annotated chip definition file (CDF) (file available on M3D, along with a summary of changes). Only probe-sets matching coding sequences were included in analysis (4217 of 7417 total probe-sets). Thus, the training dataset consisted of 4217 probe-sets by 524 arrays. The details of the test sets constructed from this compendium are presented in Table 1. For conditions sampled in replicate, the median Lasso residual values were used in ranking targets. This compendium is available for browsing and download as *E\_coli\_v4\_Build\_3* on M3D.

### 3.3 Algorithm implementation

We used the LARS implementation of Lasso regression to estimate the interaction matrix  $B$  (Efron *et al.*, 2004). A Matlab version of the LARS algorithm was downloaded from the website of Prof. Karl Sjöstrand (<http://www2.imm.dtu.dk/~kas/software/spca/index.html>). We applied 10-fold cross-validation to select the parameter  $\lambda$  (e.g. see Hastie *et al.*, 2001, Ch. 7.10). We randomized the order of groups of experiments in cross-validation, as discussed in the Results section and Supplementary Material. Since each row of  $B$  was estimated individually, it was possible to run the algorithm in parallel. Matlab scripts were compiled to executables and run on a Sun Grid Engine cluster of 94 dual-processor 2GB-RAM machines. Estimating  $B$  from the Affymetrix yeast data ( $6681 \times 647$ ) took  $\sim 4$  days using 50 nodes of the cluster.

### 3.4 Algorithms implemented for comparison

We compared the performance of our proposed SSEM-Lasso algorithm to a ‘null’ method and the previously published filter-based approach MNI (di Bernardo *et al.*, 2005). Additionally, for the simulated data, we computed target ranks by sorting the absolute value of the raw data (referred to as ‘DATA’). We compared the SSEM-Lasso method to the null and DATA methods because neither of these methods consider network information. We expect that when the network is not trivial (and enough information is provided), the SSEM-Lasso method will perform better than these methods in predicting direct targets.

The null method we used is similar to methods utilized in Dwyer *et al.*, 2007 and Tibshirani and Hastie, 2007. The  $z$ -scores were computed for each entry of the data matrix  $Y$  based on the mean and standard deviation of a gene across all experiments (a compendium-based  $z$ -score). Median  $z$ -score values were calculated for replicate samples and gene ranks were determined by sorting the absolute values of  $z$ -scores.

The MNI algorithm was implemented as described in di Bernardo *et al.*, 2005 with the exception of setting the number of metagenes (parameter  $Q$ ). For the two-color yeast compendium, we used the value of  $Q$  reported in the original publication. For the Affymetrix compendia, we set the parameter  $Q$  as recommended in Xing and Gardner, 2006: the algorithm was run over a range of values of  $Q$  for the genetic perturbation testset. We then selected the value for  $Q$  at which the normal  $z$ -score method ranked the largest fraction of tested targets in the top 100 genes. In the final round of the MNI algorithm, only 100 genes are ranked. As a consequence, target ranks were computed by MNI for each individual microarray experiment, and the median rank for

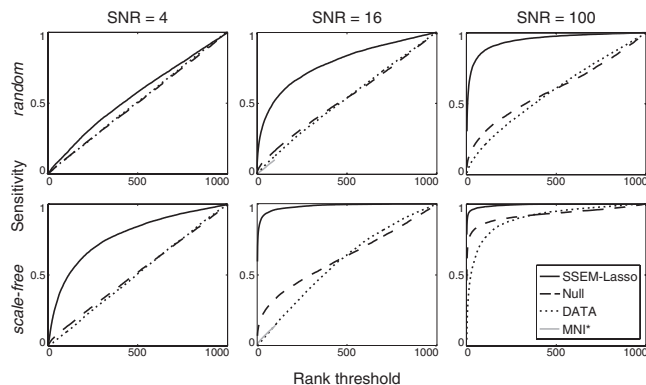
replicate experiments was reported. Unranked genes were assigned a rank of 101, which represents an optimistic strategy in the sense that it yields the best possible median MNI rank results for targets tested in replicate.

## 4 RESULTS

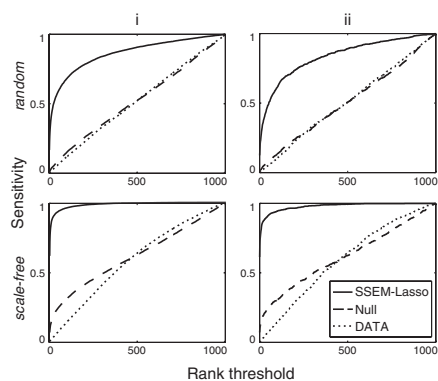
### 4.1 Simulation

We first applied our method to data generated from simulated random and scale-free networks. Initially, we generated data for estimating the interaction matrix  $B$  (training phase) by propagating only noise through the simulated network (unperturbed data). We considered these data to be closest to ideal for training the network because they represented a set of wild-type control profiles influenced only by network interactions (no outside perturbations). Then, separate test datasets in which each gene was perturbed once with SNRs of 4, 16 and 100 were used to evaluate performance in target identification. We evaluated the sensitivity of the algorithm in identifying perturbed targets at different rank thresholds, where a target was considered correctly identified at a rank threshold  $r$  if it was ranked  $\leq r$ , and sensitivity was computed as the fraction of targets correctly identified out of total targets tested. The  $r$  is analogous to the false positive rate. Performance of the MNI and null methods and the raw simulated data were also evaluated for comparison (though, we only implemented MNI for SNR = 16 because of the tuning that is required for each simulated dataset).

As can be seen in Figure 1, the SSEM-Lasso method performed the best throughout all cases. Additionally, as the SNR of the perturbations increased, the SSEM-Lasso method demonstrated larger gains in sensitivity than the other methods tested. It is also evident that at a given SNR, higher sensitivity was achieved in scale-free networks than that for random networks. This is likely attributable to a smaller average shortest path length between all pairs of nodes in scale-free networks as compared with random networks at a given sparse rate (Newman, 2003). It should be noted that we also conducted the same performance evaluation when the true interaction matrix  $B$  was used as the network filter, which corresponds to using the ideal residuals in (4) instead of those in (6). As could be expected, given the structure of the simulations,



**Fig. 1.** Plots of sensitivity versus rank threshold for all methods using simulated data generated from random (top row) and scale-free (bottom row) networks with  $p = 1000$ ,  $n = 100$  and  $sparse\ rate = 0.01$ . The SNR of applied single-gene perturbations was set to 4, 16 or 100. \*MNI only ranks the top 100 genes, and MNI was only evaluated for the SNR = 16 cases. These MNI results appear in the lower left corner of these two plots.



**Fig. 2.** Plots of sensitivity versus rank threshold for simulation results in which perturbed data were used in the training phase. The sensitivity of each method was calculated at all rank thresholds for the two test sets (i) and (ii) described in the text. This analysis was conducted for random (top row) and scale-free (bottom row) networks with  $p = 1000$ ,  $n = 100$  and sparse rate = 0.01.

we observed perfect (for SNR = 16 and 100) or near-perfect (for SNR = 4) performance (data not shown), verifying our conceptual strategy.

We repeated the analysis in Figure 1 at a lower  $B$  matrix sparse rate (sparse rate = 0.005) (Supplementary Material, Figure S1). Trends were similar to those seen in Figure 1. All methods performed better at the lower sparse rate as was expected with less network influence in these cases. The SSEM-Lasso method again performed the best throughout, with one exception: in the scale-free network high SNR case, all methods quickly achieved nearly 100% sensitivity and the null method performed best (though with <1% increase over SSEM-Lasso). This result is indicative of the fact that, at the lower sparse rate, with high SNR and the smaller average shortest path length of scale-free networks, the network information was minimal. Considering all plots in Figure 1 and Supplementary Material, Figure S1, the SSEM-Lasso method demonstrated an average 34% increase in sensitivity over the next best method over all rank thresholds, and an average 167% increase for rank thresholds  $\leq 100$ .

We also investigated the performance of the algorithm when perturbed data were used for estimating the interaction matrix  $B$ , a potentially less ideal case than the unperturbed data used in the training phase in the previous results. Accordingly, single-gene perturbation data were used to both train and test the SSEM-Lasso method. Two datasets were generated: (A) a set of 100 perturbations (100 of the 1000 total genes were perturbed once), used in the training phase for all results shown in Figure 2 and also in the test phase as test set (ii) [Fig. 2, column (ii)]; and (B) an additional set of 1000 perturbations (each gene perturbed once), used only in the test phase as test set (i) [Fig. 2, column (i)], generating results comparable to those in Figure 1, SNR = 16 column. Importantly, the use of perturbed data in the training phase appeared to have little effect on the algorithm's ability to identify target genes, as differences between performance in Figure 2 [column (i)] and Figure 1 [SNR = 16] were negligible. Additionally, only a slight drop in performance was observed when the same data were used in both the training and test phases [case (ii) compared with (i)]. These results are promising for application to real data, as the

limited amount of data available makes it preferable to include all experiments in the network training phase, even those used for testing performance [similar to case (ii)].

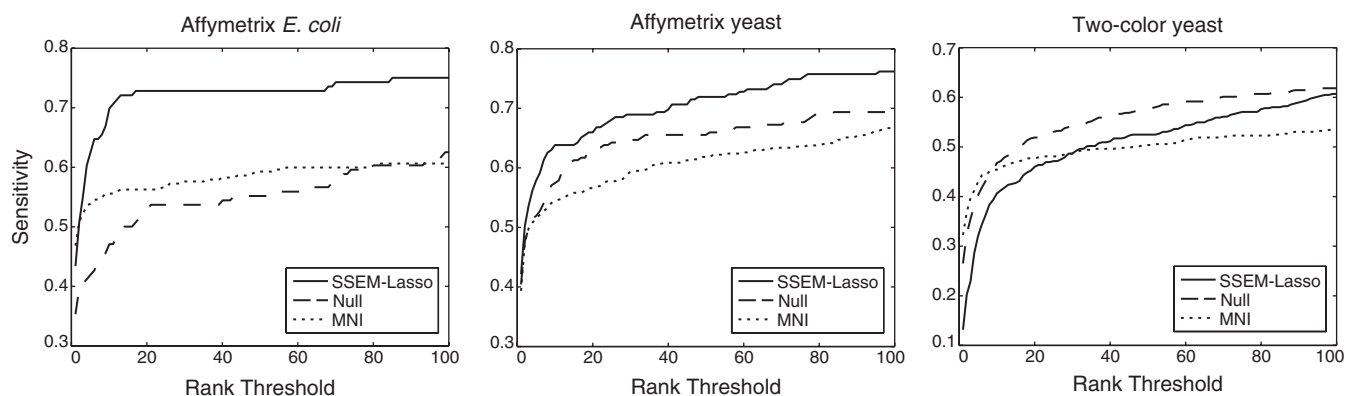
Finally, we investigated the performance of the algorithm when the perturbations targeted more than one gene (i.e. either 10 or 50), with the multiple targets selected either at random ('random') or clustered according to adjacencies in the underlying interaction matrix  $B$  ('umbrella'). As might be expected, we observed a decrease in the performance of both the SSEM-Lasso and null methods as the number of perturbed targets increased (Supplementary Material, Figure S2). Nevertheless, our algorithm still clearly outperformed the null method. Interestingly, in the case of scale-free networks, we saw that the algorithm performed better when the multiple targets were related through the interaction matrix ('umbrella') than when they were selected at random. This result is promising for application to real data, given the somewhat more representative nature of scale-free topologies for describing biological networks and the fact that multiple targets in a biological perturbation are more likely to be proximal in the network rather than dispersed randomly throughout.

## 4.2 Microarray compendia

We applied the SSEM-Lasso, MNI and null methods to three microarray compendia: a two-color yeast dataset and two Affymetrix datasets, one in yeast and one in *E. coli*, both assembled from collections of previously published studies (see Methods section). To examine the algorithm performance on real data, we tested the ability of each algorithm to identify known gene targets of genetic perturbations (i.e. gene deletion, mutation or overexpression) and drug treatments within each dataset and computed the sensitivity at different rank thresholds.

**4.2.1 Genetic perturbations** The performance of the tested algorithms in identifying the known targets of genetic perturbations is presented in Figure 3. In both Affymetrix compendia, the SSEM-Lasso method demonstrated improvement in sensitivity over the next-best method for rank thresholds  $\leq 100$ : the SSEM-Lasso method had a 24% increase over the MNI method in the *E. coli* set, and an 8% increase over the null method in the Affymetrix yeast set. This significant improvement in target identification in the Affymetrix datasets is a promising indication of the potential of the SSEM-Lasso approach in identifying the direct targets of perturbations. However, using the two-color yeast compendium, the null method performed best, with an average 9% increase in sensitivity over the SSEM-Lasso method.

We hypothesize that the poor performance of the SSEM-Lasso algorithm on the two-color yeast compendium in Figure 3 is due to a number of factors. First, the null method outperformed those that attempted to estimate and remove network effects, suggesting that network information was not aiding in identification of targets. This could be attributed to the fact that the two-color compendium is composed almost entirely of single-gene perturbations, while the Affymetrix compendia included many environmental perturbations. As suggested in Faith *et al.*, 2007a, it is possible that environmental perturbations provide more information about the underlying network interactions, improving estimation of  $B$  in the Affymetrix compendia as compared to the two-color compendium. However, we believe that the root cause of this



**Fig. 3.** Plots of sensitivity versus rank threshold for each method in identifying the target of genetic perturbations in three microarray compendia: Affymetrix *E. coli* (136 targets tested), Affymetrix yeast (235 targets tested) and two-color yeast (501 targets tested).

discrepancy is related to the inherent differences between two-color and Affymetrix microarray data; two-color data are ratios of gene expression in a treated sample to that of an internal control sample (rather than expression values), and are normalized individually within each experiment (rather than across all experiments as in RMA for Affymetrix data). We hypothesize that this use of a different internal reference in each two-color experiment (and subsequent intra-experiment normalization) leads to poor standardization and possible distortion of the signal. Indeed, for genetic perturbations in the two-color compendium, we observed a modest 3% increase in sensitivity over null with a standardization-related adjustment to our SSEM-Lasso method (Supplementary Material, Figure S3). These results support the use of single-channel microarray data for compendium-based applications involving estimation of network interactions.

As noted in the Methods section, we randomized the order of groups of experiments during cross-validation. Completely randomizing the order of the microarrays negatively affected performance; however, when groups of experiments from the same publication were maintained together in the order, randomizing the order of these groups had very little effect, as presented in detail in the Supplementary Material. We believe these results indicate that experimenter biases (e.g. due to in-lab variations in RNA sample preparation and hybridization) are present within the expression profiles, and that artifactitious effects due to these biases are lessened when groups of experiments from the same publication are removed together during cross-validation.

**4.2.2 Drug perturbations** We tested the performance of the algorithm on drug perturbations. The results of any drug target ranked <1000 by any of the three algorithms are presented in Table 2, which included 5 of 34 unique gene drug targets tested in Affymetrix *E. coli*, 4 of 18 in Affymetrix yeast and 16 of 24 in two-color yeast. In the Affymetrix compendia, overall performance was inconsistent. In both these compendia, it was difficult to compare the performance of MNI with null and SSEM-Lasso since only 100 genes were ranked by MNI, and no algorithm consistently ranked drug targets in the top 100 genes. Notably, however, the SSEM-Lasso method ranked a target of caspofungin in yeast and norfloxacin in *E. coli* in the top 100 genes, while both MNI and null

**Table 2.** Rank of known drug targets for drug treatments included in each microarray compendium

Drug	Target	Target rank		
		SSEM-Lasso	Null	MNI
<i>Affymetrix E. coli</i>				
Ampicillin	<i>dacD</i>	629	843	–
Norfloxacin, ccdB toxin	<i>gyrA</i>	82	1879	–
Kanamycin, Spectinomycin	<i>rpsA</i>	222	1642	–
Kanamycin, Spectinomycin	<i>rpsB</i>	690	2325	–
Kanamycin, Spectinomycin	<i>rpsO</i>	755	2460	–
Kanamycin, Spectinomycin	<i>rpsP</i>	912	1673	–
<i>Affymetrix yeast</i>				
Caspofungin	<i>FKS1</i>	27	215	–
Caspofungin	<i>GSC2</i>	942	3583	–
Thiolutin	<i>RPB10</i>	1494	917	–
Nocodazole, Benomyl	<i>TUB1</i>	978	865	–
<i>Two-color yeast</i>				
Cycloheximide	<i>RPL28</i>	810	92	–
Cycloheximide	<i>RPL26A/B</i>	2275	153	–
Doxycycline	<i>RPS9A</i>	460	356	–
Hydroxyurea	<i>RNR1</i>	14	10	14
Hydroxyurea	<i>RNR2</i>	20	7	6
Hydroxyurea	<i>RNR3</i>	7	23	23
Hydroxyurea	<i>RNR4</i>	4	6	2
Itraconazole	<i>ERG11</i>	17	4	2
Lovastatin	<i>HMG1</i>	89	44	98
Lovastatin	<i>HMG2</i>	31	17	30
Terbinafine	<i>ERG1</i>	1874	185	5
PTSB	<i>TRR1</i>	55	25	32
PTSB	<i>TRX2</i>	45	569	36
Dyclonine	<i>ERG2</i>	29	28	4
3-Aminotriazole	<i>CTA1</i>	325	214	–
3-Aminotriazole	<i>HIS3</i>	942	791	–

Only drug target genes ranked in the top 1000 genes by at least one method are presented in this table. The MNI method only ranks 100 genes.

failed to do so. Additionally, the SSEM-Lasso method demonstrated consistent enrichment for known drug targets compared with the null method in the *E. coli* compendium. In the two-color yeast

compendium, all three methods ranked known targets in the top 50 genes in several cases, with MNI clearly performing best in identifying the target of terbinafine. However, it is important to note that MNI was developed using this compendium, and is possibly overfit to these data (see Discussion section). Additionally, we note that the null method, based on expression  $z$ -scores alone, performed comparably to the other methods that attempted to account for network interactions in this dataset. This observation suggests that these particular drug target experiments are operating in a higher SNR regime than the drug perturbations in the Affymetrix datasets.

## 5 DISCUSSION

We have presented a statistically rigorous and supported method for identification of the direct targets of perturbations using microarray compendia. Our proposed algorithm demonstrated significant improvement over other tested methods in identifying genetic perturbation targets in the Affymetrix compendia, and showed consistent enrichment of drug targets in the Affymetrix *E. coli* compendium. Our method does not require any user-defined parameters—all tuning is done automatically—so that it can easily be applied to new datasets, as was demonstrated in its application to three different microarray compendia in this study. In this section, we present discussion on various outstanding issues relating to our method.

### 5.1 Competing methods

We note that the competing MNI algorithm was developed using the two-color yeast compendium and may be overfit to these data. In our implementation of the algorithm, we only tuned one of the six MNI parameters for each compendium individually, and set the other five to their suggested values. The poor performance of MNI on the Affymetrix compendia suggests that additional tuning of all or a subset of the other five parameters may be required. However, there are currently no optimization criteria for doing so, and such a search would be computationally demanding using these large datasets. These inconsistent results and questions of parameter tuning for MNI highlight key advantages of the SSEM–Lasso algorithm.

Our analysis also demonstrates the merits of the null method we used for comparison. The compendium  $z$ -test method we applied as a null model yielded significant improvement in sensitivity over the two null methods used in the MNI publication for two-color yeast compendium targets: an average 15% improvement over ranking expression values alone, and an average 30% improvement over ranking expression values standardized by an estimate of the technical variance of each gene probe (data not shown). Additionally, the null method performed comparably to the SSEM–Lasso and MNI methods in identifying drug targets in the two-color compendium. These results and the strong performance of the null method throughout experimental datasets tested, even compared with our proposed SSEM–Lasso method, indicate that it is a good baseline approach for looking at potential targets. It also has the appeal of requiring only a single, simple calculation once the data are assembled and appropriately normalized.

### 5.2 Choice of experimental data

There are interesting open questions regarding the optimal nature and size of a compendium for application of our proposed algorithm.

Such questions appear to be non-trivial, and a detailed investigation is beyond the scope of this article. However, we conducted a preliminary study by looking at the performance of SSEM–Lasso and null in identifying targets of genetic perturbations using different sized representative subsamples of the Affymetrix *E. coli* and yeast compendia (Supplementary Material, Figure S5). We observed different trends for each of the two compendia, making it difficult to establish rigorous guidelines for application of our algorithm to new datasets. Nevertheless, for both compendia, we did see that our proposed algorithm still demonstrated increased sensitivity in target identification compared with the null method at reduced numbers of chips. Notably, in the *E. coli* compendium, we observed a significant increase in performance over the null method with only 30 chips. Identifying which chips in a compendium are most and least informative, in order to guide experimental design and data collection, is part of our ongoing work. In general, we recommend (1) using as many chips as are available, since we never observed that adding chips hurt algorithm performance (in fact, it usually led to improvement, albeit minimal in some cases), and (2) that these chips sample as many diverse experimental conditions as possible, given that we observed a drop in performance for subsamples comprised of highly correlated chips (data not shown).

### 5.3 Predicting drug targets

Another issue clearly meriting further study is the inconsistent performance of our proposed method in identifying drug targets, a point that highlights the complexity of these perturbations. We believe these results are indicative of the multi-level nature of such perturbations, where direct interactions often take place at the protein and/or metabolite level, and therefore do not necessarily affect mRNA expression of the encoding gene(s). For example, we conducted Gene Ontology term enrichment for the top 100 ranked genes for all drug perturbations with known targets, but did not observe enrichment of the appropriate terms (data not shown), even in the cases where SSEM–Lasso ranked the known target in the top 100 (norfloxacin and caspofungin). We see three directions in which improved performance may be obtained. First, we expect that the performance of our proposed algorithm may be improved by conducting additional array experiments where the drug of interest is applied over several conditions, as was the case for norfloxacin in the *E. coli* compendium. Second, we conjecture that a more sophisticated post-processing of the SSEM–Lasso residuals, incorporating multiple data sources (e.g. Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways, signaling pathways and protein–protein interaction data, as well as any prior knowledge about the drug in question), should yield a more refined distillation of the target information contained in these residuals. For such an analysis, the use of the magnitudes of the residual effects (e.g. through an FDR analysis or similar, as commented previously), rather than just the ranks, may be desirable as well. Third, as we continue to improve our understanding of the mechanisms of action of drugs and, in particular, their impact on observed mRNA expression data, useful extensions of our model will likely present themselves. The proposed SSEM framework provides a natural foundation upon which to build such extensions.

## 5.4 Methodological extensions

As a side note, we mention that from a methodological perspective it may be of interest to explore other approaches to inferring the matrix  $B$  of network interactions. For example, it might be useful to investigate application of Lasso variations that encourage the inclusion of groups of variables, as in Ma *et al.*, 2007 or Zou and Hastie, 2005, or incorporation of an adjustment similar to that in Dobra *et al.*, 2004 to promote consistency among predicted gene interactions. Furthermore, it is appealing to try to include in the model available information pertaining to the topology of a gene regulatory network. However, some caution likely is called for in pursuing this direction, as the task of inferring a network topology is not entirely equivalent to that of predicting the effect of interactions in the network (as we do here). This distinction boils down to one of statistical variable selection versus statistical prediction, and is influenced by the presence of colinearity among the predictor variables. In the context of the Lasso, a key practical implication of this distinction is the need to tune the regularization parameter  $\lambda$  differently, depending on the goal of the analysis (for details, see Leng *et al.*, 2006 and Meinshausen and Bühlmann, 2006). We have included an expanded discussion of this distinction in the Supplementary Material, along with a comparison of our method to one developed for network topology inference.

## 5.5 Why does it work?

Overall, the proposed SSEM-Lasso method demonstrated significant improvement in sensitivity in identifying perturbation targets in simulated and real datasets. The strong performance of the SSEM-Lasso algorithm in identifying genetic perturbation targets in the Affymetrix compendia implicates the method as a very promising approach for identifying genetic mutation or dysregulation in cancer and disease. We are in the preliminary stages of applying this algorithm to an Affymetrix human cancer compendium with a collaborating laboratory.

We note, however, that in pursuing such applications, it is important that there be developed in parallel an understanding of the mathematics behind just why our methodology works as well as it does. In fact, on the face of it, that the SSEM-Lasso method should work at all may seem contrary to the conventional wisdom on such problems. Standard results in mathematics and statistics tell us that, given a regression problem of the form  $Y = X\beta + \mathcal{E}$ , where  $Y$  and  $\mathcal{E}$  are  $n \times 1$ ,  $X$  is  $n \times p$  and  $\beta$  is  $p \times 1$ , we must have  $n > p$  to expect a unique estimate of  $\beta$  based on least squares regression techniques. If  $n < p$ , the least-squares problem will be under-determined and will generally have an infinite number of solutions. Although the model in equation (3) is an auto-regressive model, the same concerns still hold, and clearly in the applications considered in this article, we have  $n < p$ .

That our proposed method still works in this context is due to an intimate connection between our problem and the newly emerged area of *compressed sensing*, at the heart of which lies precisely this type of seeming contradiction. The results in compressed sensing, developed primarily over just the past 5 years, arguably represent a revolution in applied mathematics and statistics in this regard. Specifically, results in this area for the standard regression problem state that, in the case of  $n < p$ , if (i) the unknown vector  $\beta$  is sparse, in the sense of having relatively few components of non-trivial magnitude and (ii) the matrix  $X$  has a certain technical property

(sometimes called ‘incoherence’), then it is possible to recover  $\beta$  with great accuracy. However, (i) sparse regression techniques, such as the Lasso, must be used, rather than standard least squares regression and (ii) accurate estimation of  $\beta$  typically can be assured only ‘with high probability’ in interesting problems.

Candès and Wakin, 2008 provide an excellent recent introduction to and survey of work in compressed sensing. The mathematical machinery needed to state and prove precise versions of results like that just described is rather technical, and includes techniques and tools from harmonic analysis, convex optimization and random matrix theory—the combination of which itself is a somewhat new phenomenon. This existing work does not, however, directly address the problem we consider in this article, although there are sufficient key structural similarities between our problem and those in the literature to imagine that similar results can be shown. In fact, our recent work (S.Yang and E.D.Kolaczyk, unpublished data) shows that certain sparseness-related assumptions for  $B$ , an additional assumption of a multivariate Gaussian noise  $\mathcal{E}$  (a technical convenience for ensuring ‘incoherence’) and the use of Lasso-based regression are sufficient to make mathematically precise statements about the performance that can be expected of the SSEM-Lasso method, as a function of the characteristics of the underlying network.

## ACKNOWLEDGEMENTS

We thank Melissa Dominguez for conducting YG-S98 microarray drug treatment experiments; Shu Yang for input to algorithm development; Dan Kamalic and Boris Hayete for help with the computer cluster; Ilaria Mogno for providing simulation code and general guidance and feedback; and Jeremiah Faith for general guidance and feedback.

*Funding:* National Science Foundation (NSF)/National Institutes of Health (NIH) Mathematical Biology Program (1R01GM078987-01).

*Conflict of Interest:* none declared.

## REFERENCES

- Barrett,T. *et al.* (2007) NCBI GEO: mining tens of millions of expression profiles—database and tools update. *Nucleic Acids Res.*, **35**, D760–D765.
- Bonneau,R. *et al.* (2006) The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biol.*, **7**, R36.
- Candès,E.J. and Wakin,M.B. (2008) An introduction to compressive sampling. *IEEE Signal Proc. Mag.*, **25**, 21–30.
- Courcelle,J. *et al.* (2001) Comparative gene expression profiles following UV exposure in wild-type and SOS-deficient *Escherichia coli*. *Genetics*, **158**, 41–64.
- di Bernardo,D. *et al.* (2005) Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nat. Biotechnol.*, **23**, 377–383.
- Dobra,A. *et al.* (2004) Sparse graphical models for exploring gene expression data. *J. Multivariate Anal.*, **90**, 196–212.
- Dwyer,D.J. *et al.* (2007) Gyrase inhibitors induce an oxidative damage cellular death pathway in *Escherichia coli*. *Mol. Syst. Biol.*, **3**, 91.
- Efron,B. *et al.* (2004) Least angle regression. *Ann. Stat.*, **32**, 407–451.
- Ergün,A. *et al.* (2007) A network biology approach to prostate cancer. *Mol. Syst. Biol.*, **3**, 82.
- Faith,J.J. *et al.* (2007a) Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol.*, **5**, e8.
- Faith,J.J. *et al.* (2007b) Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Res.*, doi:10.1093/nar/gkm815.

- Golub,T.R. et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.
- Hastie,T. et al. (2001) *The Elements of Statistical Learning*. Springer, New York.
- Hughes,T.R. et al. (2000) Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109–126.
- Irizarry,R.A. et al. (2003) Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res.*, **31**, e15.
- Jeong,H. et al. (2000) The large-scale organization of metabolic networks. *Nature*, **407**, 651–654.
- Jeong,H. et al. (2001) Lethality and centrality in protein networks. *Nature*, **411**, 41–42.
- Kohanski,M.A. et al. (2007) A common mechanism of cellular death induced by bactericidal antibiotics. *Cell*, **130**, 797–810.
- Leng,C. et al. (2006) A note on the Lasso and related procedures in model selection. *Stat. Sinica*, **16**, 1273–1284.
- Liao,J.C. et al. (2003) Network component analysis: reconstruction of regulatory signals in biological systems. *Proc. Natl Acad. Sci. USA*, **100**, 15522–15527.
- Ma,S. et al. (2007) Supervised group Lasso with applications to microarray data analysis. *BMC Bioinformatics*, **8**, 60.
- Marton,M.J. et al. (1998) Drug target validation and identification of secondary drug target effects using DNA microarrays. *Nat. Med.*, **4**, 1293–1301.
- Meinshausen,N. and Bühlmann,P. (2006) High-dimensional graphs and variable selection with the Lasso. *Ann. Stat.*, **34**, 1436–1432.
- Miklos,G.L. and Maleszka,R. (2004) Microarray reality checks in the context of a complex disease. *Nat. Biotechnol.*, **22**, 615–621.
- Mnaimneh,S. et al. (2004) Exploration of essential gene functions via titratable promoter alleles. *Cell*, **118**, 31–44.
- Natsoulis,G. et al. (2005) Classification of a large microarray data set: algorithm comparison and analysis of drug signatures. *Genome Res.*, **15**, 724–736.
- Newman,M.E.J. (2003) Random graphs as models of networks. In Bornholdt,S. and Schuster,H.G. (eds.) *Handbook of Graphs and Networks: From the Genome to the Internet*. Wiley-VCH, Berlin, pp. 35–68.
- Parkinson,H. et al. (2007) ArrayExpress—a public database of microarray experiments and gene expression profiles. *Nucleic Acids Res.*, **35**, D747–750.
- Ravasz,E. et al. (2002) Hierarchical organization of modularity in metabolic networks. *Science*, **297**, 1551–1555.
- Subramanian,A. et al. (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci. USA*, **102**, 15545–15550.
- Tibshirani,R. (1996) Regression shrinkage and selection via the Lasso. *J. Roy. Stat. Soc. B Met.*, **58**, 267–288.
- Tibshirani,R. and Hastie,T. (2007) Outlier sums for differential gene expression analysis. *Biostatistics*, **8**, 2–8.
- Xing,H. and Gardner,T.S. (2006) The mode-of-action by network identification (MNI) algorithm: a network biology approach for molecular target identification. *Nat. Protoc.*, **1**, 2551–2554.
- Zou,H. and Hastie,T. (2005) Regularization and variable selection via the elastic net. *J. Roy. Stat. Soc. B Stat. Met.*, **67**, 301–320.