Published online 19 February 2010 *Nucleic Acids Research, 2010, Vol. 38, No. 11* ***3523–3532***
doi:10.1093/nar/gkq045

GOing Bayesian: model-based gene set analysis of genome-scale data

Sebastian Bauer[1], Julien Gagneur[2] and Peter N. Robinson[1,3,4,*]

[1]Institute for Medical Genetics, Charité-Universitätsmedizin Berlin, Augustenburger Platz 1, 13353 Berlin, [2]European Molecular Biology Laboratory, Meyerhofstrasse 1, 69117 Heidelberg, [3]Max Planck Institute for Molecular Genetics, Ihnestrasse 73, 14195 Berlin and [4]Berlin-Brandenburg Center for Regenerative Therapies (BCRT), Charité-Universitätsmedizin Berlin, Berlin, Germany

Received November 30, 2009; Revised January 14, 2010; Accepted January 15, 2010

ABSTRACT

The interpretation of data-driven experiments in genomics often involves a search for biological categories that are enriched for the responder genes identified by the experiments. However, knowledge bases such as the Gene Ontology (GO) contain hundreds or thousands of categories with very high overlap between categories. Thus, enrichment analysis performed on one category at a time frequently returns large numbers of correlated categories, leaving the choice of the most relevant ones to the user's interpretation.

Here we present model-based gene set analysis (MGSA) that analyzes all categories at once by embedding them in a Bayesian network, in which gene response is modeled as a function of the activation of biological categories. Probabilistic inference is used to identify the active categories. The Bayesian modeling approach naturally takes category overlap into account and avoids the need for multiple testing corrections met in single-category enrichment analysis. On simulated data, MGSA identifies active categories with up to 95% precision at a recall of 20% for moderate settings of noise, leading to a 10-fold precision improvement over single-category statistical enrichment analysis. Application to a gene expression data set in yeast demonstrates that the method provides high-level, summarized views of core biological processes and correctly eliminates confounding associations.

INTRODUCTION

Many studies in functional genomics follow a data-driven approach. Experiments such as transcriptional profiling with microarrays, ChIP-on-chip or gene knock-out screens are done at the scale of the whole genome without specifying a prior hypothesis. Instead, one seeks to discover new phenomena and generate new hypotheses from the data. Loosely formulated, the main question driving the analysis of data-driven experiments is: *what is going on?*

Although for technical and biological reasons the nature of the data differs between these types of experiments, they often can be summarized by a list of genes which responded to the experiment, e.g. genes found to be differentially expressed, bound by a particular transcription factor or whose knock-down elicits a phenotype of interest. However, extensive lists of responder genes are not *per se* useful to describe the experiment. A practical way to address the question of *what is going on?* is to perform a gene category analysis, i.e. to ask whether these responder genes (which we will refer to as the *study set*) share some biological features that distinguish them among the set of all genes tested in the experiment (which we will refer to as the *population*). Gene category analysis involves a list of gene categories, such as those provided by the Gene Ontology (GO) (1) or the pathways of the KEGG database (2), and a statistical method for identifying enriched categories such as overrepresentation analysis using Fisher's exact test (3), gene set enrichment (4–7), logistic regression (8), random-set analysis (9) and Bayesian techniques for analyzing GO terms in a context where not all annotated genes are observed (10).

Gene category analyses that follow the mentioned approaches often return a large number of significant categories, which are related to one another and leave to the user the task of choosing the most meaningful categories at the risk of relying on biased judgments. The reason for the correlation is that genes can belong to multiple categories, so that if one category is significantly overrepresented in the study set, then it is more likely that other categories with many genes in common with it will also be significantly overrepresented. While

*To whom correspondence should be addressed. Tel: +49 30 450569122; Fax: +49 30 450569915; Email: peter.robinson@charite.de

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

these enrichments are correct in a statistical sense, long lists of results can make it difficult to focus on the most important ones in a biological sense.

With GO, annotations are understood to follow the *true-path rule*, meaning that whenever a gene is annotated to a term it is also implicitly associated with all parents of that term. This is an important cause of gene sharing, a phenomenon which we have termed the *inheritance problem* (11). A number of methods have been proposed to deal with these problems. The elim algorithm examines the GO graph in a bottom-up fashion (12). Once a GO term is found to be significantly overrepresented according to Fisher's exact test, all genes annotated to it are removed from further analysis. A variant of this algorithm downweights the genes rather than eliminating them completely (12). The parent–child algorithm determines overrepresentation of a term in the context of annotations to the term's parents by redefining the groups used to calculate the hypergeometric distribution (11).

All of the aforementioned procedures successively test overrepresentation for each of the categories. The above methods make use of the structure of GO to address statistical dependencies, but they are limited to ontologies and do not fundamentally differ from the original paradigm of term-for-term testing with the exact Fisher test. More recently, an approach called GenGO was presented, which fits a model on all the terms simultaneously. GenGO models the set of responder genes as members of a set of active GO terms. The objective of GenGO is to identify the most likely combination of active terms, while allowing for some false positive and some false negative responder genes. The fitting procedure optimizes an objective function that combines the likelihood of the model with a penalization that tends to limit the overall number of active terms. This procedure was shown to outperform other methods for detecting GO-term overrepresentation on simulated data and to identify concise yet biologically relevant sets of significantly overrepresented GO terms when applied to real data sets (13).

Here we present model-based gene set analysis (MGSA), a model-based approach that significantly improves over GenGO by scoring terms with their posterior probabilities, leading to more robust results and to improvements over standard methods across a broader range of sensitivity cutoffs. Our benchmark on simulated data confirm the drastic improvements of model-based approaches over term-by-term methods. MGSA uses a simple and adaptable Bayesian network that provides a great deal more flexibility as to the kinds of data that can be used in the analysis and to extensions of the model. MGSA is implemented as part of the free and open-source software Ontologizer (14), a Java application that implements a large number of GO enrichment methods and enables visual exploration of the results.

METHODS

Model

We model gene response in a genome-wide experiment as the result of an activation of a number of biological categories. These categories can be pathways as defined by the KEGG database (2), GO terms (15) or any other scheme (5,16) that associates genes to potentially overlapping biologically meaningful categories. Because we primarily work with GO, we call these categories *terms*. Our method does not make use of the graph structure of GO other than utilizing the true-path rule, which states that if a gene is associated to a term, then it is also associated to all of terms along the path up to the root of the ontology. Apart from that we make no explicit use of the structure.

We assume that the experiment attempts to detect genes that have a particular *state* (such as differential expression), which can be *on* or *off*. The true state of any gene is hidden. The experiment and its associated analysis provide observations of the gene states that are associated with unknown false positive (α) and false negative rates (β), which we will assume to be identical and independent for all genes.

For instance, in the setting of a microarray experiment, the *on* state would correspond to differential expression, and the *off* state would correspond to a lack of differential expression of a gene. Our model, hence, assumes that differential expression is the consequence of the annotation to some terms that are *on*.

An additional parameter p represents the prior probability of a term being in the *on* state. The probability p is typically low (<0.5), introducing an effective penalization for the number of active terms. This ingredient promotes parsimonious explanations of the data.

More formally, our model is a Bayesian network with three layers augmented with a set of parameters (Figure 1):

(1) The *term layer* $T = \{T_1, \ldots, T_m\}$ consists of Boolean nodes that represent the m terms. There is a Boolean variable associated with each node that can have the state values *on* (1) or *off* (0).
(2) The *hidden layer* $H = \{H_1, \ldots, H_n\}$ contains Boolean nodes that represent the n annotated genes. There are edges from the terms to their annotated genes. For instance, if gene H_1 is annotated to terms T_1 and T_2

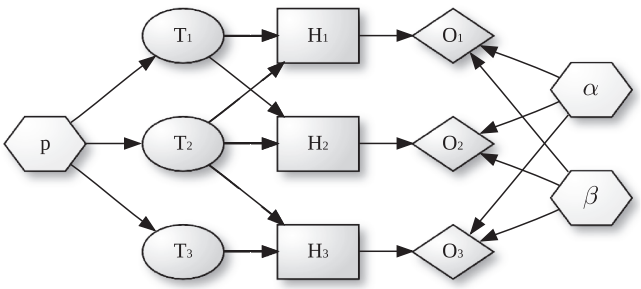

Figure 1. A Bayesian network to model gene response with gene categories. Gene categories, or terms (T_i, ellipses) can be either *on* or *off*. Terms that are *on* activate the hidden state (H_j, rectangles) of all genes annotated to them, the other genes remain *off*. The observed states (O_j, diamonds) of the genes are noisy observations of their true hidden state. The parameters of the model (light gray nodes) are the prior probability of each term to be active, p, the false positive rate, α and the false negative rate, β.

then there is an edge between T_1 and H_1 and another edge between T_2 and H_1. The state of the nodes reflects the true activation pattern of the genes. Each node can have the state values *on* (1), or *off* (0).

(3) The *observed layer* $O = \{O_1, \ldots, O_n\}$ contains Boolean nodes reflecting the state of all observed genes. The observed gene state nodes are directly connected to the corresponding hidden gene state nodes in a one-to-one fashion.

(4) The *parameter set* contains continuous nodes with values in [0,1] corresponding to the parameters of the model α, β and p. These parameterize the distributions of the observed and the term layer as detailed below.

For didactic purposes, we will initially explain a simplified version of our procedure in which the parameters α, β and p are considered to have known, fixed values. We will then show how the Bayesian network can be augmented to search for optimal values for α, β and p.

The state propagation of the nodes can be modeled using various *local probability distributions* (LPDs), denoted by P. The joint probability distribution for this Bayesian network can be written as

$$P(T,H,O) = P(T)P(H|T)P(O|H) = P(T)\prod_{i=1}^{n} P(H_i|T)P(O_i|H_i). \quad (1)$$

We model the state of each $T_j \in T$ according to a Bernoulli distribution with hyperparameter p, i.e. $P(T_j = 1) = p$. Denote by $m_{x|T}$ the number of terms that have state x for a given T, i.e. $m_{x|T} = |\{j \,|\, T_j = x\}|$ then

$$P(T) = p^{m_{1|T}}(1-p)^{m_{0|T}}. \quad (2)$$

In the following, we denote by $T(H_i) \subseteq T$ the set of terms to which gene H_i is annotated, i.e. H_i and the ancestors of H_i. For the $T \rightarrow H$ links, any node $H_i \in H$ is *on* ($H_i = 1$) if at least one term in $T(H_i)$ is *on*. Otherwise it is *off*:

$$P(H_i = 1|T) = \begin{cases} 1 & \text{if } \exists\, T_j \in T(H_i) : T_j = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Note that this transition is deterministic. For the $H \rightarrow O$ connection, we choose the following two Bernoulli distributions: $P(O_i = 1|H_i = 0) = \alpha$ and $P(O_i = 0|H_i = 1) = \beta$.

Therefore, α is the probability that a gene i is observed to be *on* (i.e. $O_i = 1$), although its true hidden state is actually *off* (i.e. $H_i = 0$) and thus, none of the terms which annotate the gene are *on*. Correspondingly, β is the probability of a gene being observed to be *off*, although at least one term that annotates it is *on*.

Denote by $n_{xy|T} = |\{i|O_i = x \wedge H_i = y\}|$ the number of genes having observed activation x and true activation y according to the states of T. For instance, $n_{01|T}$ corresponds to the number of genes observed to be not differentially expressed but whose true activation state is *on*. Then, by considering the LPDs of nodes, we get the following product of Bernoulli distributions for $P(O|T) = \prod_{i=1}^{n} P(H_i|T)P(O_i|H_i)$:

$$P(O|T) = \alpha^{n_{10|T}}(1-\alpha)^{n_{00|T}}(1-\beta)^{n_{11|T}}\beta^{n_{01|T}}. \quad (4)$$

Markov Chain Monte Carlo algorithm for marginal probabilities inference with known parameters

As it is often the case, marginal posteriors for our network cannot be derived analytically. We estimate these values using a Metropolis–Hasting algorithm, which is a Markov chain Monte Carlo (MCMC) method (17–19). The MCMC algorithm performs a random walk over the term and parameter configurations, which asymptotically provides a random sampler according to the target distribution $P(T|O)$.

Given the current configuration of the terms denoted by T^t, the algorithm proposes a neighbor state T^p according to a proposal density function $Q_T(\cdot|T^t)$. We sample a value r uniformly from the range (0,1). Then, if

$$r < P_{accept}(T^t, T^p) = \frac{P(T^p|O)Q_T(T^t|T^p)}{P(T^t|O)Q_T(T^p|T^t)} \quad (5)$$

the proposal is accepted, i.e. $T^{t+1} = T^p$, otherwise it is rejected, i.e. $T^{t+1} = T^t$. Using Bayes' law, we have

$$P(T^p|O) = \frac{P(O|T^p)P(T^p)}{P(O)} \quad (6)$$

and similarly for T^t. Substituting these expressions for $P(T^p|O)$ and $P(T^t|O)$ cancels out the normalization constant $P(O)$. The acceptance probability is then:

$$P_{accept}(T^t, T^p) = \frac{P(O|T^p)P(T^p)Q_T(T^t|T^p)}{P(O|T^t)P(T^t)Q_T(T^p|T^t)}. \quad (7)$$

Equation (5) is used iteratively to define a random walk through the space of configurations. A *burn-in period* consisting of a certain number of iterations is used to initialize the MCMC chain (in our implementation, the default is 20 000 iterations). Following this, l further iterations (default 10^6) are performed. Let $C(T_i)$ be the number of samples in which term T_i was *on*. Then

$$P(T_i|O) \approx \frac{C(T_i)}{l}.$$

In order to finish the description of the algorithm, we need to define classes of operations of which a proposal is chosen, that is, we need to specify $Q_T(T^p|T^t)$. We denote by $T^p \leftrightarrow_T T^t$ the binary relation that states that T^p be constructed from T^t by either

- toggling the *on*/*off* state of a single term, or by
- exchanging the state of a pair of terms that contains a single *on* term and a single *off* term.

We denote by $N(T)$ the *neighborhood* of a given configuration for T, that is, the number of different operations that can be applied once to T in order to get a new configuration. At first, there are m terms in total, each of which can be toggled. In addition, there are $m_{0|T}m_{1|T}$ possibilities to combine terms that are *on* with terms

that are *off*. Thus, there are a total of $N(T) = m + m_{0|T}m_{1|T}$ valid state transitions. We would like to sample the valid proposals with equal probability, therefore the proposal distribution Q_T is determined by

$$Q_T(T^p|T^t) = \begin{cases} \frac{1}{N(T^t)} & \text{if } T^p \leftrightarrow_T T^t \\ 0 & \text{otherwise.} \end{cases}, \quad (8)$$

which we can use to rewrite Equation (7) to:

$$P_{accept}(T^t,T^p) = \frac{P(O|T^p)P(T^p)N(T^t)}{P(O|T^t)P(T^t)N(T^p)}.$$

The procedure is shown in Algorithm 1. For simplicity, the burn-in period is omitted from the pseudocode. It is easy to see that all states of the chain are reachable from any state, as the Markov chain is finite and it is possible to reach an arbitrary state from any other state by a fixed number of operations. This accounts for the *reducibility* of the chain. Moreover, the chain is *aperodic* as it is always possible to stay in the same state, as any proposal can be rejected. Therefore, the resulting Markov chain is *ergodic*, which is a sufficient condition for a convergence to a *stationary* distribution, which matches the desired target distribution.

Algorithm 1: A Metropolis–Hasting algorithm to estimate $P(T_1 = 1|O)$.

Data: O, l (number of steps)
Result: $P(T_1 = 1|O), \ldots, P(T_m = 1|O))$

$T^t \leftarrow \underbrace{(0,\ldots,0)}_{m \text{ times}}$;

for $t \leftarrow 1$ **to** l **do**
  $T^p \sim Q_T(\cdot|T^t)$, i.e. choose a neighbor candidate by either
    - toggling a term
    - exchanging an active term with an inactive one

  $a \leftarrow \frac{P(O|T^p)P(T^p)N(T^t)}{P(O|T^t)P(T^t)N(T^p)}$

  $r \sim U(0,1)$

  if $r < a$ **then**
    $T^t \leftarrow T^p$
  end
end

return $\left(\frac{C(T_1)}{l}, \ldots, \frac{C(T_m)}{l}\right)$

MCMC algorithm for marginal probabilities inferrence with unknown parameters

The estimation of the parameter α, β and p can be easily integrated directly into the MCMC algorithm. The parameters now must be explicitly considered in the joint probability distribution:

$$P(p,T,H,\alpha,\beta,O) = P(p)P(T|p)P(H|T)P(\alpha)P(\beta)P(O|H,\alpha,\beta), \quad (9)$$

where $P(T|p)$ is given by Equation (2), $P(H|T)$ is given by Equation (3) and $P(O|H,\alpha,\beta)$ corresponds to $P(O|H)$ of the basic model. As p, α and β are now true random variables, we must define a prior distribution on them as well. Here, we have used uniform distributions to introduce as little bias as possible.

We are seeking for a scheme to sample from joint posterior distribution

$$P(p,T,\alpha,\beta|O) = \frac{P(p,T,\alpha,\beta,O)}{P(O)}.$$

In order to utilize the Metropolis–Hasting algorithm for this purpose, we are required to provide an efficient calculation for the numerator. This is straighforward, because the numerator factors to

$$P(p,T,\alpha,\beta,O) = P(p)P(T|p)P(\alpha)P(\beta)P(O|T,\alpha,\beta), \quad (10)$$

and moreover, $P(O|T,\alpha,\beta)$ can be determined using Equation (4).

In addition to term state transitions, we also need to take parameter transitions within the proposal density into account. We define the new proposal density as a mixture of the state transition density Q_T and a parameter transition density Q_Θ. We denote the current realization of the parameters by $\Theta^t = \{\alpha^t,\beta^t,p^t\}$ and by $\Theta^p \leftrightarrow_\Theta \Theta^t$ the relation of whether Θ^p can be constructed from Θ^t. The fully specified proposal density is then

$$Q_s(T^p,\Theta^p|T^t,\Theta^t) = \begin{cases} Q_T(T^p|T^t)s & \text{if } T^p \leftrightarrow_T T^t \text{ and } \Theta^p = \Theta^t \\ Q_\Theta(\Theta^p|\Theta^t)(1-s) & \text{if } \Theta^p \leftrightarrow_\Theta \Theta^t \text{ and } T^p = T^t \\ 0 & \text{otherwise.} \end{cases}$$

The parameter $s \in (0,1)$ can be used to balance state transition proposals against parameter proposals. That is to say, depending on the outcome of a Bernoulli process with hyperparameter s, we either propose a new state transition or a new parameter setting. For the experiments described in this article, s was set to 0.5.

Many possibilities for the proposal density of the parameter transition Q_Θ and for the relation $\leftrightarrow_\Theta$ can be envisaged. We have considered transitions $\Theta_p \leftrightarrow_\Theta \Theta_p$ for which Θ_p differs from Θ_t in the realization of not more than a single variable.

In contrast with the configuration space of the terms' activation state, the domain of these new variables is continuous. However, an internal study revealed that the algorithm is not overly sensitive to the exact parameter settings. Therefore, we can restrict the range of the

variables to a set of discrete values. For the experiments described in this work, we used the restrictions $\alpha,\beta \in \{0.05k | 0 < k < 20\}$ and $p \in \{1/m,...,20/m\}$, where m is the number of terms.

At last, we can state the proposal density function for parameter transitions

$$Q_\Theta(\Theta^p|\Theta^t) = \begin{cases} \frac{1}{|A|+|B|+|P|} & \text{if } \Theta^p \leftrightarrow_\Theta \Theta^t \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

in which A, B and P stand for the domain of the parameters α, β and p respectively. Note that Q_Θ is symmetric, *i.e.* $Q_\Theta(\Theta^t|\Theta^p) = Q_\Theta(\Theta^p|\Theta^t)$.

Simulation and evaluation of the performance

The simulations were based on revision 1.846 (dated October 21, 2009) of the GO term definition file. We restricted the entire simulation study to genes of *Drosophila melanogaster*. Annotations for this species were taken from revision 1.157 (dated October 19, 2009) of the gene association file provided by FlyBase (20), using all annotations regardless of their evidence code. This results in a total of 12 484 genes that are annotated directly or by inheritance to 7078 GO terms.

Study sets were generated according to our model as follows. One value for the false positive rate α and one for the false negative rate β were set. A number (varying from one to five) of unrelated terms (i.e. pairs of terms related by parent–child relationships were avoided) are randomly picked to be in *on* state. In the remainder of this section, we denote by l_{ij} the state or label of term i within study set j, i.e. $l_{ij} = 1$, if term i is *on*, or $l_{ij} = 0$ otherwise.

Each single study set j is then filled with all genes that are annotated to the term T_i for all $l_{ij} = 1$. Next, the noisy observations are simulated by removing each gene with a probability of β from the study set. Then, genes from the population not annotated to any of the active GO terms were added to the study set with a probability of α. The whole procedure was repeated 1500 times for each combination of α and β providing 1500 different study sets of varying sizes for that combination.

All tested algorithms were then applied to the simulated study sets. Note that the study set generation procedure controls merely the expected values of the proportion of false positive and false negative genes for the study sets, whereas the actual proportion of each individual study set may differ. MGSA′ and GenGO′ (which use fixed values of the parameters α, β and p) were supplied with the values of α and β used for the simulation and p was set according to the number of GO terms that were set to *on*. The application of the algorithms results in prediction values (scores) for l_{ij}, denoted by p_{ij}. We remark that for posterior marginal probabilities higher values (rather than lower as with P-values) indicate stronger support for the state *on*.

Benchmarking of the methods was done by using standard measures for the evaluation of discrimination procedures. We made use of receiver operating characteristic (ROC) curves and precisison/recall curves, pooling the results of all study sets with identical parameter combinations. In addition to the values of a ROC analysis, we calculated the k-truncated ROC value for each study set j via

$$\mathrm{ROC}_k(j) = \frac{1}{kP}\sum_{i=1}^{k} t_i,$$

in which $P = \sum_i l_{ij}$ is the total number of positives and t_i represents the number of true positives above the i-th false positive (21,22). We reported the average over k-truncated ROC values of all study sets for $k = 10$.

Analysis of yeast growth media

Raw tiling array data comparing yeast fermentative growth (YPD: Yeast extract Pepton Dextrose) and respiratory growth (YPE: Yeast extract Peptone Ethanol) (23) were processed to provide normalized intensity values for each probe in each hybridization. The expression level of each transcript in each growth condition was estimated by the midpoint of the *shorth* (shortest interval covering half of the values) of the probe intensities of the transcript across all arrays of the growth condition. Transcripts were called expressed if their expression level was above the threshold (24). Transcript expression levels of the two conditions 'YPD' and 'YPE' were normalized against each other using the *vsn* method (25) as differential expression at the transcript level appeared to still depend on average expression value. Next, transcripts were called differentially expressed if they showed at least 2-fold change between the two conditions. We then compared a study set of 510 differentially expressed genes to the population of 5308 genes (23) (Supplementary Table 1). We used GO annotations obtained from the *Saccharomyces* Genome Database (26) as of October 22, 2009 and restricted our analysis to the *biological process* ontology.

RESULTS

Bayesian networks to model experimental observations

In order to summarize the meaning of a long list of genes by naming biologically meaningful categories or terms, we propose a knowledge-based system in form of a Bayesian network. We model the state of the genes as a function of the activity of the associated terms. The true state of the genes is hidden and propagated to corresponding entities at the observation layer, by which we reflect the noisy nature of the data. The errors between the observation and the hidden states are assumed to be independent and to occur with a potentially unknown false positive (α) and false negative rate (β), identical for all genes. Furthermore, an additional parameter p represents the prior probability of a term being in the *on* state (Figure 1). The purpose of this model is to infer the marginal posterior probability of each term i being active, i.e. $P(T_i|O)$, given the observations of the experiment. See the 'Methods' section for a formal introduction of the model and the inference process.

In practice, the parameters α, β and p are not known. While the estimation is taken into account by the specification of the algorithm, we also run instances in which these parameters are fixed *a priori* to the true ones. This provides an upper bound for the parameter estimation. In the following, we refer to the general case as MGSA and to the case with known parameters as MGSA′.

Performance on simulated data

We simulated 1500 study sets in which the number of active terms varied from one to five ('Methods' section). The simulations were performed with 12 484 *Drosophila* genes that are annotated directly or indirectly via to parent–child relationships to 7078 GO terms. We followed this approach for each combination of $\alpha \in \{0.1, 0.4\}$ and $\beta \in \{0.25, 0.4\}$, resulting in a total of 6000 simulated study sets. We then compared MGSA and MGSA′ against three single-term association procedures: the standard term-for-term (TfT) GO overrepresentation analysis by Fisher's exact test (3), Parent–Child union (PCU) analysis (11) and topological weight (TopW) analysis (12), and against the other global model approach, GenGO (13). Similar to our approach, GenGO has two parameters that are intended to capture false positive and false negative responders and an additional parameter that penalizes superfluous terms. In the original implementation of GenGO (13), a heuristic procedure was used to search for the best values of these parameters. Unfortunately, the full GenGO software is not applicable for batched runs. We have implemented the algorithm denoted as GenGO′ in the simple case where the parameters are known. For the simulations described here, we follow the authors' recommendation to set the penalty parameter to 3, while the remaining parameters were set to the optimal values. This provides an upper bound on the performance of the GenGO procedure with unknown parameters.

GO analyses typically contain a very large number of terms. Therefore, an important issue is whether a GO analysis method inflates the number of terms reported as significant. The most critical measure is therefore the precision, i.e. the proportion of true positives among the true and false positives. Comparing the precision of the different methods as a function of the recall, which is the proportion of true positives among all positive terms, demonstrates the drastic improvement of global model approaches. Both global model methods, GenGO′ and MGSA, dominate all three single-term association approaches by a factor of at least 3 (5 for MGSA) in precision at 20% recall (Figure 2A, B) across all investigated parameter settings. For a false positive rate

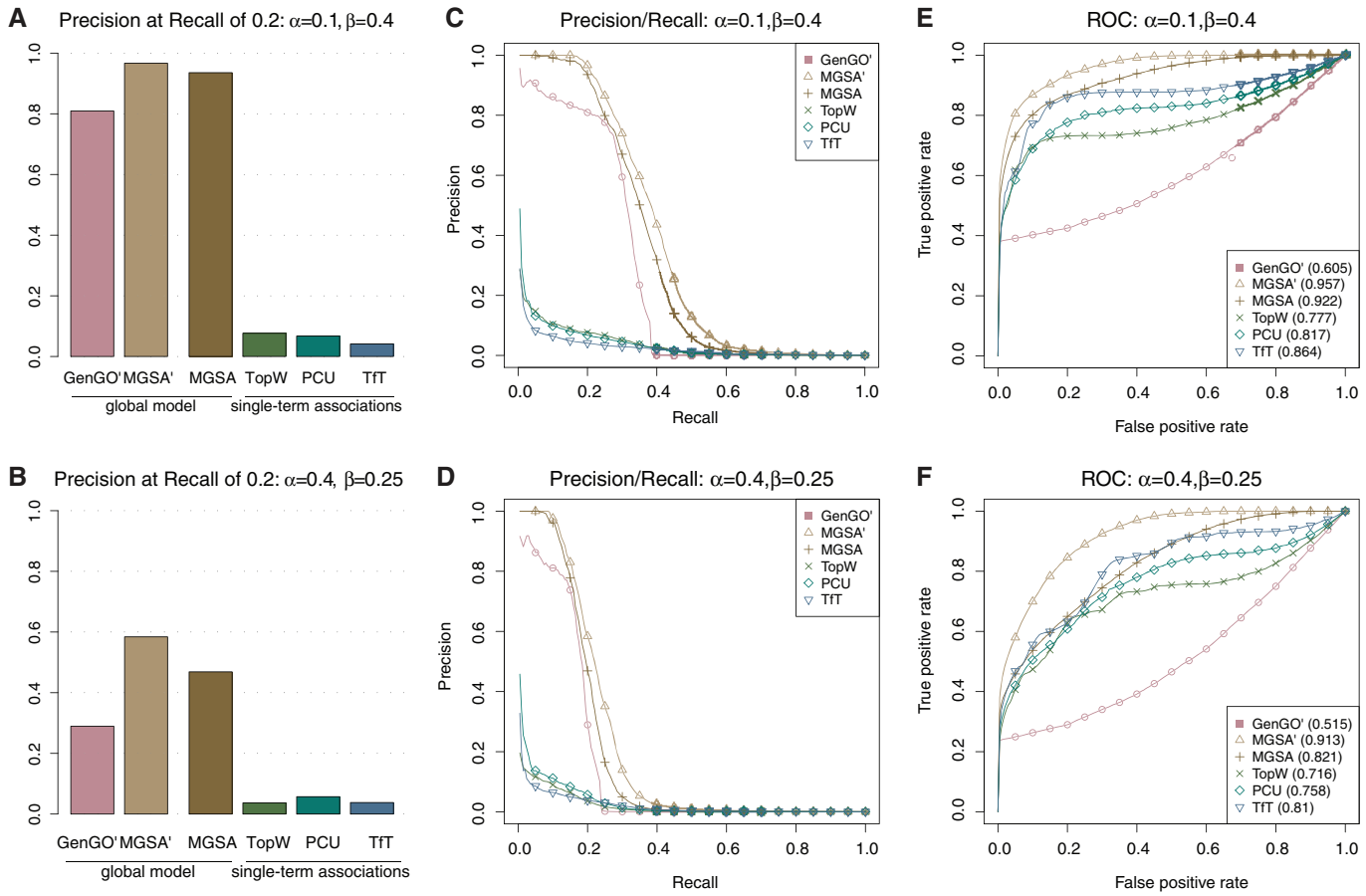

Figure 2. Benchmarking on simulated data set. Performance of the TfT, PCU, TopW, GenGO′, MGSA′ and MGSA algorithms on simulated data set with different settings of false positive (α) and false negative (β) rates. In each row, the leftmost panel shows the precision for a recall of 0.2 (**A**, **B**), the middle panel precision as a function of recall (**C**, **D**) and the rightmost panel the ROC curve (**E**, **F**).

Table 1. ROC_{10} Analysis

α	β	GenGO′	MGSA′	MGSA	TfT	PCU	TopW
0.1	0.4	0.35	0.44	0.41	0.31	0.24	0.26
0.4	0.25	0.22	0.28	0.25	0.22	0.17	0.15

k-truncated ROC curves were generated for the simulated data shown in Figure 2 for $k = 10$. The ROC10 score is the area under the ROC curve up to the tenth false positive. *k*-truncated ROC scores range from 0 to 1, with 1 corresponding to the most sensitive and selective result.

α of 0.1, the improvement reaches even 8- to 10-fold. Moreover, MGSA largely outperforms GenGO' in all settings, for example, with a precision of ~95% versus ~80% for GenGO' in the case of $\alpha = 0.1$ and $\beta = 0.4$. Values of *k*-truncated ROC scores ('Methods' section and Table 1) confirm the ranking of these methods when focusing on stringent cutoffs. Moreover, these improvements of MGSA are seen at any cutoff. MGSA outperform all other methods for the whole range of recall cutoffs and with all investigated parameter settings (Figure 2C, D).

Notably, the performance of GenGO′, which reports only a single maximum likelihood solution and discards any alternative solution, even if it is almost as likely, drops off much earlier than MGSA (Figure 2C, D). This behavior is more apparent in Receiver Operating Characteristic curves (ROC curves, Figure 2E, F), which plot the true positive rate (or recall) as a function of the false positive rate (proportion of false positives among all negative terms). Indeed, away from the most stringent zone, GenGO appears as the least accurate of all tested methods. See also Supplementary Figures S1–S12 for results on other parameter combinations

Together these results on simulation confirm the drastic improvement of global model approaches and demonstrate that our marginal posterior method, MGSA, largely outperforms GenGO by showing an accurate behavior on the whole range of cutoffs.

Dealing with unknown values of the parameters α, β and p had required a significant extension of our basic algorithm ('Methods' section). The simulation data allowed us to investigate the ability of the full MGSA algorithm to cope with unknown parameter values. We ran the basic version of the algorithm, MGSA′, in which the parameters are known and fixed *a priori*. Performance of MGSA′ are displayed in the precision–recall and the ROC curves (Figure 2) and represent what MGSA could reach if the parameters were known. As Figure 2 shows, the performance of the algorithm is not drastically affected by this, showing that the full MGSA algorithm performs reasonably well when dealing with unknown parameter values.

Analysis of expression data from fermentative and respiratory respiration in yeast

We applied our method to 510 yeast genes found to be differentially expressed in fermentative growth compared with respiratory growth ('Methods' section). Respiration and fermentation are two well-studied growth modes of yeast, thus facilitating the interpretation of the results. We ran TfT, PCU and MGSA on the *biological process* ontology of GO. Using Benjamini–Hochberg correction for multiple testing and a cutoff of false discovery rate of 0.1, the term-for-term method returns 193 significantly enriched terms, many of which are highly related (Supplementary Table 1). We investigated how the results of MGSA fluctuate by running 20 independent Markov chains, each of length 10^7, using a cutoff of 0.5 on the posterior probability to call a term *on*, i.e. a level at which a term is estimated to be more likely to be *on* than to be *off*. MGSA reports only seven terms with a marginal posterior probability >0.5 (in at least one of the 20 runs). Out of these 7 terms, 6 showed a posterior probability >0.5 consistently across the 20 chains (Figure 3 and Supplementary Table 1), while the seventh one, *monocarboxylic acid metabolic process*, had estimated posterior probability between 0.410 and 0.502. Hence, results for the most likely terms were reproducible between runs. We checked the robustness of these results against variations in the study set by creating 2 000 random subsamples of the study set containing 90% of the original genes. The terms identified by the original analysis were consistently identified in the subsamples (Supplementary Figure 13).

Among the seven terms, *oxidation reduction* summarizes the main biological process that distinguishes growth in these two different media, namely the use of oxidation phosphorylation during respiration to regenerate ATP. The other terms, such as, *carbohydrate transport*, or *monocarboxylic acid metabolic* process capture processes that are linked to the change of carbon source but not directly involved in the oxidation reduction reactions. Hence, MGSA provides a high-level, summarized view of the core biological process, respiration, avoiding redundant results while still keeping the necessary level of granularity in other branches of the ontology.

The term *cell death* illustrates very well the difference between single-term association approaches and global model approaches. Both tested enrichment-based approaches, TfT and PCU, report *cell death* as an enriched term whereas MGSA does not. It happens that mitochondria are implicated both in cell death and in respiration (27). The differentially expressed genes annotated to *cell death* encode mitochondrial proteins and are also involved in respiration. Hence, it is correct to report *cell death* enriched for differentially expressed genes. However, cells are not dying in any of these two conditions. The enrichment is due to the sharing of genes with respiration, a process which is genuinely differentially activated. In this study set, 113 genes are annotated to *oxidation reduction* including 25 out of 28 genes annotated to *cell death*. MGSA, which infers the terms that are on and not simply enriched, does not report cell death. One should also note that cell death is not a type of respiratory pathway or vice versa. Methods such as TfT that examine the statistical significance of each term separately cannot compensate for correlations between terms due to gene sharing. Although methods such as PCU and TopW can compensate for some kinds of statistical correlations that arise because of the inheritance of annotations from descendent nodes in the GO graph (11,12), they fail in

A term-for-term

GO ID	Name	NSP	P-Value	Adj. P-Value	Rank
GO:0055114	oxidation reduction	B	.80e-41	2.16e-38	1
GO:0006091	generation of precursor metabolites and e	B	.39e-32	1.43e-29	2
GO:0015980	energy derivation by oxidation of organic c	B	.32e-26	1.43e-23	3
GO:0045333	cellular respiration	B	.76e-26	1.43e-23	4
GO:0006119	oxidative phosphorylation	B	.35e-23	2.24e-20	5
GO:0009060	aerobic respiration	B	.62e-19	5.24e-17	6
GO:0006811	ion transport	B	.58e-19	1.13e-16	7
GO:0006084	acetyl-CoA metabolic process	B	.71e-18	1.01e-15	8
GO:0006099	tricarboxylic acid cycle	B	.44e-17	5.34e-15	9
GO:0046356	acetyl-CoA catabolic process	B	.44e-17	5.34e-15	10
GO:0008219	cell death	B	.99e-16	2.00e-14	11
GO:0016265	death	B	.99e-16	2.00e-14	12
GO:0051187	cofactor catabolic process	B	.97e-16	3.67e-14	13
GO:0009266	response to temperature stimulus	B	.04e-16	6.04e-14	14
GO:0042773	ATP synthesis coupled electron transport	B	.20e-15	8.45e-14	15
⋮					
GO:0044247	cellular polysaccharide catabolic process	B	0.0132	0.0830	189
GO:0006083	acetate metabolic process	B	0.0132	0.0830	190
GO:0006089	lactate metabolic process	B	0.0132	0.0830	191
GO:0006817	phosphate transport	B	0.0147	0.0922	192
GO:0006826	iron ion transport	B	0.0153	0.0954	193

B MGSA

GO ID	Name	NSP..	Marginal	Rank
GO:0055114	oxidation reduction	B	1.00	1
GO:0009266	response to temperature stimulus	B	0.975	2
GO:0009636	response to toxin	B	0.861	3
GO:0015891	siderophore transport	B	0.814	4
GO:0006820	anion transport	B	0.779	5
GO:0008643	carbohydrate transport	B	0.574	6
GO:0032787	monocarboxylic acid metabolic process	B	0.393	7
GO:0009250	glucan biosynthetic process	B	0.391	8
GO:0009310	amine catabolic process	B	0.348	9
GO:0051099	positive regulation of binding	B	0.339	10

C

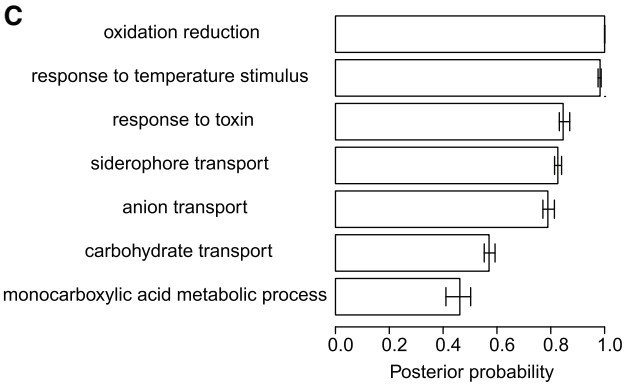

Figure 3. Application on a respiratory versus fermentative growth expression dataset in yeast. (**A**) Ranked list of the 192 overrepresented terms using a term-for-term Fisher's test with Benjamini–Hochberg correction for multiple testing. Many of the top terms are redundant and relate to similar functions. The term cell death (highlighted in blue) is a spurious association (see text). (**B**) Ranked list of the top 10 terms identified by a single run of MGSA (six of them with a posterior >0.5 in green). (**C**) Error bars (95% confidence intervals) obtained with 20 runs of MGSA. Each of the seven terms was identified with a posterior >0.5 in at least one of the 20 runs.

situations such as the one described here because, *oxidation reduction* and *cell death* share some annotated genes but are not directly connected to one another by the graph structure of GO.

DISCUSSION

Of trees and forests

Data-driven molecular biology experiments can be used to identify a list of genes that respond in the context of a given experiment. With the advent of technologies, such as microarray hybridization and next-generation sequencing, which enable biologists to generate data reflecting the response profiles of thousands of genes or proteins, gene category analysis has become ever more important as a means of understanding the salient features of such experiments and for generating new hypotheses. By using the knowledge as it is provided by GO, KEGG or other similar systems of categorization, these analyses have become a *de facto* standard for molecular biological research. Almost all previous methods are based on algorithms that analyze each term in isolation. For each term under consideration, the methods consider whether the study set is significantly enriched in genes annotated to the term compared with what one would expect based on the frequency of annotations to the term in the entire population of genes or using other related statistical models (28).

We suggest that single-term association methods that determine the significance of each term in isolation essentially do 'not see the forest for the trees', by which we mean that they tend to return many related terms which are statistically significant if considered individually, but they are not designed to return a set of core terms that together best *explain* the set of genes in the study set. Although some methods have been developed that partially compensate for statistical dependencies in GO (11,12), the work of Lu and colleagues (13) addressed the problem by *modeling* the gene responses using all categories *together*.

Modeling requires formulating a generative process of the data. We, and Lu and colleagues (13), considered the categories as the potential cause of the gene responses. Fitting the model then enables distinguishing between the causal categories (according to the model) from the categories merely associated with gene response. Although one cannot conclude that the identified categories are causal in reality (this is only a model and one only has observational data), this feature of model-fitting explains why it provides a better answer to the question *what is going on?* than testing for associations on a term for term basis.

Searching for an optimal set of terms that together explain a biological observation is a more difficult problem than examining each term for enrichment one at a time. In particular, the model used to find term sets must specify how the terms interact with one another. This imposes

assumptions on the model that must be kept in mind when the results are interpreted. For instance, the model presented in this work assumes that activation of a single term suffices to activate genes, and does not require that a certain minimum number of genes are annotated by the term set. The Bayesian framework we present can easily be adapted for different kinds of models encoding different biological assumptions by choosing different priors or distributions. This will be the subject of future research.

A Bayesian framework for inference

In contrast to the method of Lu and coworkers (13), we embed the model into a Bayesian Network, to which we apply standard methods of probabilistic inference. This not only leads to an intuitive derivation of the score, but also increases the versatility of the framework. That is, although we have demonstrated our method with simple classes of LPDs for the nodes, one can easily use more involved distributions. Moreover, we show that by augmenting the Bayesian network, we get a streamlined approach that includes both the inference of the states and of the parameters. This in principle also enables the inclusion of prior knowledge, be it parameters that are known or estimated in form of $P(\alpha)$ and $P(\beta)$, or a specific term configuration in form of $P(T)$ or $P(T|p)$.

Instead of scoring terms by finding the maximum *a posteriori* as in the GenGO method (13), we use marginal posterior probabilities. Finding the maximum *a posteriori* provides a single combination of active terms and is not informative about alternative solutions. If several solutions show near-maximal likelihood then it is implausible that the single one with the largest likelihood is always the right solution. In contrast, marginal posterior as in MGSA associates a natural weight to each term that reflects a measure of certainty of its involvement in the process. Importantly, using marginal posterior probabilities increases the robustness and lowers the sensitivity of the procedure to high variance related to the multimodality of the problem, i.e. the existence of local maxima of the likelihood function. As we demonstrated using simulations, MGSA is indeed more robust than GenGO.

The Java implementation that we provide in the Ontologizer estimates the marginal posteriors with a MCMC algorithm and is fast enough to perform 10^6 steps in <3 s on a standard 2.5 GHz PC. Since runs of MCMC are not guaranteed to converge in any *a priori* defined number of steps, we suggest that users repeat the analysis in order to see how the reported marginal probabilities of the top terms fluctuate. If fluctuations are too large, the number of MCMC steps should be increased.

Using a Bayesian approach that models the data with all categories simultaneously, rather than using hypothesis testing on each category, avoids the issue of multiple testing. Moreover, the interpretation of the score, which is simply the probability of a category to be active, might appear more natural than a *P*-value. Finally, one should note that the ranking of the scores is reversed to values given by hypothesis-based procedures, i.e. high marginals give high support in the Bayesian setting, while high confidence in the hypothesis-based approach is indicated by low *P*-values.

CONCLUSION

We have addressed the question of gene category analysis using a model-based approach, MGSA. In this Bayesian model, the genes responding to the experiment are assumed to belong to a small number of 'active' categories. Therefore, to answer the question of *what is going on* in an experiment, MGSA infers the 'active' categories, among all considered categories, given the actual gene state observations. We have shown that under the assumptions of our model our approach is better in identifying the causal sets than other procedures. We suggest that considering the forest instead of the trees is an advantageous strategy for gene category analysis, and that global model procedures such as the one presented in this article may be better able to describe the biological meaning of high-throughput data sets than are procedures that examine associations of categories one at a time. We note that the Bayesian network analyzed in this article is but one of many potential network structures that are made possible with our framework.

Our methods have been integrated into the *Ontologizer* project, an easy-to-use Java Webstart application for performing analysis of overrepresentation. The Ontologizer as well as the implementation of the described benchmark procedure have been released under the terms of the modified BSD licence. The application is available from http://compbio.charite.de/index.php/ontologizer2.html. The source code is available from http://sourceforge.net/projects/ontologizer/.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We are grateful to Emilie Fritsch for insightful discussions about the yeast gene expression dataset analysis and to Simon Anders for critical reading of the manuscript.

FUNDING

This work was supported by grants from the Deutsche Forschungsgemeinschaft (DFG SFB 760 and RO 2005/4-1). Additional support was provided by the Lab of Lars Steinmetz. Funding for open access charge: Deutsche Forschungsgemeinschaft.

Conflict of interest statement. None declared.

REFERENCES

1. The Gene Ontology Consortium. (2000) Gene Ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.
2. Kanehisa,M. and Goto,S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, **28**, 27–30.
3. Rhee,S.Y., Wood,V., Dolinski,K. and Draghici,S. (2008) Use and misuse of the Gene Ontology annotations. *Nat. Rev. Genet.*, **9**, 509–515.
4. Jiang,Z. and Gentleman,R. (2007) Extensions to gene set enrichment. *Bioinformatics*, **23**, 306–313.
5. Mootha,V.K., Lindgren,C.M., Eriksson,K.-F., Subramanian,A., Sihag,S., Lehar,J., Puigserver,P., Carlsson,E., Ridderstrle,M., Laurila,E. *et al.* (2003) PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.*, **34**, 267–273.
6. Nam,D. and Kim,S.-Y. (2008) Gene-set approach for expression pattern analysis. *Brief. Bioinform.*, **9**, 189–197.
7. Oron,A.P., Jiang,Z. and Gentleman,R. (2008) Gene set enrichment analysis using linear models and diagnostics. *Bioinformatics*, **24**, 2586–2591.
8. Sartor,M.A., Leikauf,G.D. and Medvedovic,M. (2009) LRpath: a logistic regression approach for identifying enriched biological groups in gene expression data. *Bioinformatics*, **25**, 211–217.
9. Newton,M.A., Quintana,F.A., den Boon,J.A., Sengupta,S. and Ahlquist,P. (2007) Random-set methods identify distinct aspects of the enrichment signal in gene-set analysis. *Ann. Appl. Stat.*, **1**, 85–106.
10. Vêncio,R.Z.N., Koide,T., Gomes,S.L. and Pereira,C.A.B. (2006) BayGO: Bayesian analysis of ontology term enrichment in microarray data. *BMC Bioinformatics*, **7**, 86.
11. Grossmann,S., Bauer,S., Robinson,P.N. and Vingron,M. (2007) Improved detection of overrepresentation of Gene-Ontology annotations with parent child analysis. *Bioinformatics*, **23**, 3024–3031.
12. Alexa,A., Rahnenführer,J. and Lengauer,T. (2006) Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*, **22**, 1600–1607.
13. Lu,Y., Rosenfeld,R., Simon,I., Nau,G.J. and Bar-Joseph,Z. (2008) A probabilistic generative model for GO enrichment analysis. *Nucleic Acids Res.*, **36**, e109.
14. Bauer,S., Grossmann,S., Vingron,M. and Robinson,P.N. (2008) Ontologizer 2.0—a multifunctional, tool for GO term enrichment analysis and data exploration. *Bioinformatics*, **24**, 1650–1651.
15. Barrell,D., Dimmer,E., Huntley,R.P., Binns,D., O'Donovan,C. and Apweiler,R. (2009) The GOA database in 2009 – an integrated Gene Ontology Annotation resource. *Nucleic Acids Res.*, **37**, D396–D403.
16. Subramanian,A., Tamayo,P., Mootha,V.K., Mukherjee,S., Ebert,B.L., Gillette,M.A., Paulovich,A., Pomeroy,S.L., Golub,T.R., Lander,E.S. *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.
17. Andrieu,C., De Freitas,N., Doucet,A. and Jordan,M.I. (2003) An introduction to MCMC for machine learning. *Mach. Learn.*, **50**, 5–43.
18. Diaconis,P. (2009) The Markov chain Monte Carlo revolution. *Bull. Am. Math. Soc.*, **46**, 179–205.
19. Diaconis,P. and Saloff-Coste,L. (1995) What do we know about the Metropolis algorithm?. STOC'95: *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*. *ACM* pp. 112–129.
20. Tweedie,S., Ashburner,M., Falls,K., Leyland,P., Mcquilton,P., Marygold,S., Millburn,G., Osumi-Sutherland,D., Schroeder,A., Seal,R. *et al.* (2009) FlyBase: enhancing Drosophila Gene Ontology annotations. *Nucleic Acids Res.*, **37(Suppl. 1)**, D555–D559.
21. Gribskov,M. and Robinson,N.L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem.*, **20**, 25–33.
22. Schaffer,A.A., Aravind,L., Madden,T.L., Shavirin,S., Spouge,J.L., Wolf,Y.I., Koonin,E.V. and Altschul,S.F. (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.*, **29**, 2994–3005.
23. Xu,Z., Wei,W., Gagneur,J., Perocchi,F., Clauder-Münster,S., Camblong,J., Guffanti,E., Stutz,F., Huber,W. and Steinmetz,L.M. (2009) Bidirectional promoters generate pervasive transcription in yeast. *Nature*, **457**, 1033–1037.
24. David,L., Huber,W., Granovskaia,M., Toedling,J., Palm,C.J., Bofkin,L., Jones,T., Davis,R.W. and Steinmetz,L.M. (2006) A high-resolution map of transcription in the yeast genome. *Proc. Natl Acad. Sci. USA*, **103**, 5320–5325.
25. Huber,W., von Heydebreck,A., Sültmann,H., Poustka,A. and Vingron,M. (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, **18(Suppl. 1)**, S96–S104.
26. Hong,E.L., Balakrishnan,R., Dong,Q., Christie,K.R., Park,J., Binkley,G., Costanzo,M.C., Dwight,S.S., Engel,S.R., Fisk,D.G. *et al.* (2008) Gene Ontology annotations at SGD: new data sources and annotation methods. *Nucleic Acids Res.*, **36**, D577.
27. Green,D.R. and Kroemer,G. (2004) The pathophysiology of mitochondrial cell death. *Science*, **305**, 626–629.
28. Goeman,J.J. and Bühlmann,P.P. (2007) Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics*, **23**, 980–987.