Inference of Gene Relations from Microarray Data by Abduction

Irene Papatheodorou, Antonis Kakas*, and Marek Sergot

Department of Computing, Imperial College London, SW7 2AZ, UK
{ivp,ack,mjs}@doc.ic.ac.uk

Abstract. We describe an application of Abductive Logic Programming (ALP) to the analysis of an important class of DNA microarray experiments. These experiments measure differences in expression levels of whole genomes in differing environmental conditions and/or after deletion or overexpression of one or more genes. Their aim is to obtain insights about gene interactions and gene pathways. We develop an ALP theory that provides a simple and general model of how gene interactions can cause changes in observable expression levels of genes. Input to the procedure are the observed microarray results; output are hypotheses about possible gene interactions that explain the observed effects. A key feature of the model are parameters that encode different biological assumptions and provide a means of constraining the search for possible hypotheses. We have applied and evaluated our approach on microarray experiments on *M.tuberculosis* and on *S.cerevisiae* (yeast). Comparison of inferred hypotheses against known gene regulation networks and known gene functions in the biological literature provide a form of independent validation of the model.

1 Introduction

In recent years the focus in bioinformatics has shifted from the analysis of genome sequences, now available in their entirety for several organisms, to functional genomics, which broadly described seeks to ascribe biological function to genes, groups of genes and other genome features, and to understand gene interactions. One of the most important tools in these studies is DNA microarray technology. DNA microarrays measure expression levels of thousands of genes simultaneously. One common form of experiment aims at identifying genes whose expression is affected by environmental conditions or by changes in expression of other genes. The aim is to obtain clues about gene interactions and unravel pathways that define the cell's responses to various stimuli. The datasets generated by such experiments are too large and complex for manual analysis. The raw data are analysed using statistical techniques to establish which genes have been significantly differentially expressed. Methods for further interpretation of the results,

* Visiting from Department of Computer Science, University of Cyprus.

in terms of identifying gene interactions and pathways, remain largely undeveloped, however, though Bayesian Networks have recently attracted attention (see e.g. [2]).

In this paper we formulate the analysis of this type of microarray data as a problem of *abduction*, that is, inference from observable effects, in this case the microarray gene expression data, to possible causes, explanatory hypotheses about possible gene interactions that would account for the observable effects. We construct an Abductive Logic Program (ALP) theory which provides a simple, general model of how gene interactions can cause changes in observable expression levels of genes—essentially a formalisation of the (usually implicit) reasoning used by biologists when designing microarray experiments of this type. It is clear that there may be exponential numbers of compatible hypotheses in the general case, in which every gene is possibly related to other genes. A key feature of the model are adjustable parameters that allow us to constrain the search for possible hypotheses and apply the methods to realistically large data sets. The gene relations abduced are then further processed to construct possible interaction networks.

The methods are being developed and evaluated on microarray experiments on *Mycobacterium tuberculosis* and *S. cerevisiae* (yeast). The model is validated by comparing the explanatory hypotheses generated against known gene interactions in these organisms and by assessing the biological plausibility of the hypotheses where detailed information is lacking. These hypotheses in turn are intended to help our biological collaborators design a new series of microarray experiments. Section 3 presents three examples of inferences obtained from the *M. tuberculosis* datasets, including one which re-discovers part of the heat shock response pathway reported in the biological literature. At this stage the validation of the model is still fragmentary. Currently, we are testing the model on microarray data on yeast, an organism for which there are large amounts of publicly available experimental data and for which gene interactions are comparatively well understood and documented.

To avoid any misunderstanding, we want to state explicitly that we are aware there are a great many issues, and other experimental techniques, in the search for gene regulation mechanisms. We are restricting attention here to one possible form of analysis, of one of many types of experimental data. The emphasis in this paper is on the ALP model and its implementation.

2 The Model

The microarray experiments we use can be divided into four groups. They compare expression levels of:

1. A mutant sample against the wild type;
2. A sample exposed to an environmental condition against the wild type;
3. A mutant sample exposed to an environmental condition against the wild type;
4. A mutant sample exposed to an environmental condition against another (usually wild type) sample exposed to the same environmental condition.

Gene mutation involves either gene inactivation (e.g. by knock out) or over-expression (rendering a gene constantly active). The environmental change can be either deprivation of a compound/nutrient or exposure of the organism to extreme conditions, e.g. heat shock, or to abundance of a compound/nutrient.

Input to the abduction procedure is a collection of observations expressed as logic assertions of the form *increases_expression(Expt, Gene)* and *reduces_expression(Expt, Gene)*. These are obtained by a (standard) statistical analysis of the raw microarray data to determine the significance of measured differences of expression levels of each gene (see e.g. [6]). *increases_expression(Expt, Gene)* and *reduces_expression(Expt, Gene)* represent that the statistical significance of differentiated expression of gene *Gene* in experiment *Expt* exceeds a specified threshold. The value of this threshold can be adjusted.

The output from the procedure is a set of abducible relations of two different types: *induces(Gene1,Gene2)* and *inhibits(Gene1,Gene2)* for the hypothesis that *Gene1* induces the expression of *Gene2*, or inhibits it, respectively. Each individual experiment provides partial clues about possible *induces*/*inhibits* relations between genes. Unlike e.g. the methods based on Bayesian Networks referred to earlier, which analyse one experiment at a time, our abductive method seeks sets of explanatory hypotheses that account for the combined observations of many experiments together, of all four types in the list above. Some examples are provided in the later sections.

2.1 The ALP Framework

The modelling framework we employ is Abductive Logic Programming (ALP) [3, 9], an extension of logic programming that allows declarative logical representations of the problem domain and supports abductive reasoning. In ALP, a theory is represented by a triple (P, A, IC), where P is a logic program, A a set of abducible predicates and IC is a set of classical logic formulae, the *integrity constraints*.

Definition 1. *Given an abductive logic theory (P, A, IC), an abductive explanation for a query (observation) Q is a set Δ of ground abducible atoms on the predicates A such that:*

- $P \cup \Delta \models_{LP} Q$
- $P \cup \Delta$ *is consistent*
- $P \cup \Delta \models_{LP} IC$.

where $\models_{LP}$ denotes some standard logical entailment relation of logic programming.

The abductive explanation Δ represents a hypothesis, which together with the model described in program P explains how a nonempty experimental observable Q could hold. In practice abductive explanations are also required to satisfy some minimality and other evaluation criteria. In this application we do not require that the abductive explanations are minimal. Non-minimal hypotheses may still be of some biological interest.

The role of the integrity constraints IC is to impose additional *validity* requirements on the hypotheses Δ. They are modularly stated in the theory, in addition to the basic model captured in P. They are used to augment any partial information on the abducible predicates or to impose other constraints on the abductively generated explanations. They can also be used, as explained below, to steer the search for specific forms of explanation that our domain experts are looking for.

2.2 Gene interactions

Top-level Rules The program P of the ALP theory represents how gene interactions can increase or reduce the expression of genes, as observed in the experiments. An assumption is that such observed variations in gene expression should be attributed directly or indirectly to the variations (gene mutations or environmental stress), carried out in the experiment(s) investigated.

For example: if an experiment E knocks out a gene G, and G inhibits gene X, then E will show an increased expression of X — subject to some possible exceptions. This rule is expressed in logic programming notation as follows:

$$\begin{aligned} &\textit{increases\_expression(E, X)} \leftarrow \\ &\quad \textit{knocks\_out(E, G), inhibits(G, X),} \\ &\quad \textit{not incr\_affected\_by\_other\_gene(E,G,X),} \\ &\quad \textit{not incr\_affected\_by\_EnvFact(E, X).} \end{aligned} \tag{1}$$

E is a variable that ranges over names of experiments and G, X are variables that represent genes. *increases_expression(E,X)* is observational data from the experiment E, *inhibits(G,X)* is part of the unknown information to be abduced, and *knocks_out(E,G)* provides background knowledge about the experiment E.

The last two conditions of rule (1) express possible exceptions: it could be that (a) the expression of gene X is affected by a gene other than G which cancels out the effect of G on X, or (b) the expression of gene X is affected by an environmental factor (e.g. heat shock, or nutrient deprivation) of the experiment E. Here *not* is the logic programming construct 'negation as failure', used to express that (1) is a default general rule subject to the stated exceptions. The treatment of exceptions in the abductive process is discussed below.

Similarly, the following rule deals with the cases of reduced expression of G in experiment E:

$$\begin{aligned} &\textit{reduces\_expression(E, X)} \leftarrow \\ &\quad \textit{knocks\_out(E, G), induces(G, X),} \\ &\quad \textit{not red\_affected\_by\_other\_gene(E,G,X),} \\ &\quad \textit{not red\_affected\_by\_EnvFact(E, X).} \end{aligned} \tag{2}$$

There are similar rules that cover the cases of over-expressing G in the experiment E, and other rules that deal with the various combinations of gene mutation and changes in environmental conditions identified in the classification of experiment types listed in the previous section.

The rules (1) and (2) only account for direct relationships between the mutated gene G and the differentially expressed gene X. These relationships could be indirect: it might be that G regulates X *via* some intermediary gene Gx. Inference of intermediate steps of interaction is accommodated by the addition of further rules containing recursive steps, as follows:

$$\begin{aligned} &\textit{increases\_expression(E, X)} \leftarrow \\ &\quad \textit{mutates(E, G), intermediary\_gene(E, Gx, G),} \\ &\quad \textit{reduces\_expression(E, Gx), inhibits(Gx, X),} \\ &\quad \textit{not incr\_affected\_by\_other\_gene(E,Gx,X),} \\ &\quad \textit{not incr\_affected\_by\_EnvFact(E, X).} \end{aligned} \tag{3}$$

If gene Gx inhibits gene X, and the expression of gene Gx is reduced (directly or indirectly) by the mutation (knock out or over-expression) of gene G in experiment E, then the expression of X is increased in the experiment E. The relation *mutates(E,G)* covers both knock-out and over-expression of gene G in the experiment E. The relation *intermediary_gene(E, Gx, G)* is one of the parameters of the model, as discussed below.

The Exceptions: The exceptions deal with the possibility that the difference in gene expression can be attributed to a factor other than the mutated gene. The first exception in (1) is captured by the relation *incr_affected_by_other_gene(E,G,X)*, defined as follows:

$$\begin{aligned} &\textit{incr\_affected\_by\_other\_gene(E,G,X)} \leftarrow \\ &\quad \textit{increases\_expression(E, Gx),} \\ &\quad Gx \neq X,\ Gx \neq G, \\ &\quad \textit{related\_genes(Gx, G), induces(Gx, X)} \end{aligned} \tag{4}$$

$$\begin{aligned} &\textit{incr\_affected\_by\_other\_gene(E,G,X)} \leftarrow \\ &\quad \textit{reduces\_expression(E, Gx),} \\ &\quad Gx \neq X,\ Gx \neq G, \\ &\quad \textit{related\_genes(Gx, G), inhibits(Gx, X).} \end{aligned} \tag{5}$$

Rule (4) expresses the possibility that some gene Gx (other than G), whose expression is observed to increase in the experiment E could induce gene X and therefore cancel out the effect of the knocked out gene G on gene X. In that case we would not necessarily expect to see an increase in expression of X in experiment E, even if G does inhibit X. Rule (5) deals with the case where another gene Gx, whose expression is reduced in experiment E, inhibits gene X and therefore cancels out the effects on gene X of knocking out the inhibiting gene G in the experiment.

Both rules (4) and (5) refer to the abducible relations *induces* and *inhibits*. The abductive procedure will search for combinations of *induces* and *inhibits* that explain the observed increase/reduction of expression levels in an experiment whilst taking into account these exceptions.

The second exception in (1) is captured by *incr_affected_by_EnvFact(E,X)*. This relation is completely defined, in the sense that it does not depend on the abducible relations. It holds when the increase in expression of X can be

attributed to an environmental factor in experiment E. It is evaluated by comparing the expression level of the gene to its level in another experiment where a wild-type sample has been exposed to the same environmental stress.

There are several implementations of the ALP framework available. We employ a modified version of the implementation obtainable from [4]. This has the feature that abductive hypotheses and negation as failure literals *not* B, which computationally are treated as abducibles, are evaluated only when they are (can be made) ground. In this case if B is a consistent hypothesis (i.e., B fails in the program P extended with the final set of hypotheses Δ), then *not* B is included as part of the explanatory hypothesis generated (as opposed to attempting to further decompose the *not* B hypothesis into constituent parts). This simple treatment of *not* B is appropriate for the representation of exceptions to default general rules required in this application; more general forms of abduction and abductive computation (see e.g. [**?**,1]) could be employed but do not seem to add significant extra value in this application.

The Parameters: The rules that define the first exception contain the *parametric* condition *related_genes(Gx, G)*. This condition selects the genes we take into account when searching for hypotheses.

Parameters such as this are used to reduce the space of possible hypotheses. For example, the relation *related_genes(Gx, G)* can be defined in such a way that *Gx* is 'related' to *G* when gene *Gx* is a regulator of known similar function to *G*. By varying the definition of the relation *related_genes* we can explore and test different possibilities of the model. (There is no loss of generality because *related_genes(Gx, G)* can also be defined to hold for all genes *Gx* and *G*.)

The relation *intermediary_gene(E, Gx, G)* appearing in rule (3) is another example. Again, its definition can vary, to generate and test different forms of explanatory hypotheses. A simple definition specifies that any regulator gene *Gx* except for the mutant gene *G* of the experiment *E* is a candidate intermediary gene. We can further restrict the search for candidate intermediary genes by formulating integrity constraints that consider only those regulator genes that are affected (increased or reduced expression) in the experiment *E*.

We cannot over-emphasis the importance of these parameters. In the general case there may be an exponential number of possible hypotheses. The parameters allow us to constrain the problem.

2.3 Validity requirements

The integrity constraints IC express the validity requirements imposed on the abducible relations. We form constraints of three different types.

(1) self-consistency: For example, a gene cannot both inhibit and induce the same gene at the same time (under the same conditions):

$$\leftarrow \textit{induces(G1,G2), inhibits(G1,G2).} \tag{6}$$

(We write all integrity constraints in clausal form, with all variables implicitly universally quantified as usual.)

(2) consistency with background information: For example, we may know that a certain gene g does not have an inhibitory effect on any gene. We express this background knowledge by constraints of the form:

$\leftarrow$ *inhibits(g,G).*

In bacterial genomes, an *operon* is a group of genes that reside next to each other along the same DNA strand and are expressed together on the same mRNA. Usually, the products of these genes take part in the same biological processes, but have different functions. It can be assumed therefore that two different genes *G1* and *G2* of the same operon cannot both induce the same gene *X*, expressed as an integrity constraint as follows:

$\leftarrow$ *induces(G1,X), induces(G2,X), same_operon(G1,G2).*

There is also a similar constraint for the *inhibits* relation. The relation *same_operon* is computed from known information on gene function and location on the genome that we add as further background knowledge to the model.

(3) experimental consistency: When analysing the results of an experiment *E* in which a gene *G* is mutated, we may want to consider as 'intermediary genes' only genes whose expression is also observed to be affected in experiment *E*. This restriction is expressed by the integrity constraint:

affects(E, Gx) $\leftarrow$ *intermediary_gene(E, Gx, G), mutates(E,G).*

where *affects(E, Gx)* is defined as the disjunction of *increases_expression(E, Gx)* and *decreases_expression(E, Gx)*.

This is an optional integrity constraint. These provide a third type of parameter for the model. Validity requirements are modular and can easily be changed without requiring further restructuring of the model. Different aspects of the model can be tested easily by changing the integrity constraints. For example, the integrity constraint (6) is not always appropriate, since it is possible that *G1* induces *G2* in one set of experimental conditions but inhibits *G2* in another.

2.4 Inference of Paths

The model, as described above, outputs a set of binary gene relations in the form *induces/2, inhibits/2*. These relations are not transitive: if *G1* induces *G2* and *G2* induces *G3*, this does not necessarily mean that *G1* induces *G3*. In all but the simplest organisms, whether a gene *G1* induces/inhibits a gene *G2* may depend on how *G1* itself was expressed. It may be that *G1* induces *G2* when *G1* is induced by a gene *G3* but not when it is induced by another gene *G3′*. The inferred *induces, inhibits* relations can be composed into gene interaction paths, but these paths must be consistent with what is observed in the microarray experiments.

We define a gene path as the longest loop-free chain of the form $R_1(g_0, g_1), R_2(g_1, g_2), \ldots, R_n(g_{n-1}, g_n)$ where each R_i is either *induces* or *inhibits* and for every g_i,

apart from g_n, there is an experiment E that (i) mutates (knocks out or overexpresses) g_i, and (ii) either increases or reduces expression of every downstream element $R_{i+1}(g_i, g_{i+1}), \ldots, R_n(g_{n-1}, g_n)$.

The model can be formulated so that the hypothess generated are gene paths rather than *induces/inhibits* relations. However, since the *induces/inhibits* hypotheses already provide valuable insights, we prefer to leave the model unchanged and construct gene paths from *induces/inhibits* hypotheses in a separate, and optional, post-processing phase. The path generation program: (1) selects all terminal elements of the paths, (2) recursively propagates the paths by selecting every upstream element that is consistent with the path definition, and (3) removes any sub-chains from the set of valid paths. Sub-chains are paths that appear as fragments of longer paths.

Similarly, the raw *induces/inhibits* hypotheses are subjected to a further (optional) post-processing stage where we look for cases where several regulatory genes are needed together to activate a target gene.

3 Application: *M. tuberculosis*

The methods described in this paper are being developed and evaluated using datasets from experiments on *M. tuberculosis* obtained from two sources: our collaborators at the Centre for Microbiology and Infection (CMMI) at Imperial College London [6], and the publicly available tables from the Schoolnik lab at Stanford University [10].

Datasets from 14 microarray experiments containing approximately 300 genes each are stored in a relational database together with information from the Sanger Centre [11] about the sequenced strain *M. tuberculosis* H37Rv including predicted gene product, gene length, links to other resources such as GenBank, and a standard classification of known gene function. We also have information from the Institut Pasteur about transcriptional regulators.

In order to perform an analysis of these experimental data, selected portions of the database, usually covering observations from several different types of experiment together, are converted into logic programming notation and input to the ALP system. We have also developed a set of visualisation tools for displaying both experimental data and inferred hypotheses in the form of directed graphs, constructed using a modified version of the graph-layout software in the open source *Graphviz* package from AT&T Laboratories [14]. A web-based front-end to explore and manipulate the graphical displays is also available [5].

The examples below demonstrate the functionality of our model. In each case the parameters *related_genes* and *intermediary_gene* were defined to restrict attention to possible interactions between 16 genes of known regulatory function.

A simple observation For a first illustration, we attempt to explain a single observation from a single experiment. The observation *increases_expression(hspR,Rv0350)* represents that the difference of expression levels of gene Rv0350 in experiment 'hspR' exceeded the specified significance threshold. 'hspR' is an experiment in which a mutant sample with gene Rv0353/hspR knocked out is compared against

the wild type. The output is a hypothetical explanation, presented in the form of a list:

Hyp = [inhibits(Rv0353,Rv0350),
not(induces(Rv0352,Rv0350)),
not(incr_affected_by_other_gene(hspR,Rv0350,Rv0353)),
not(affected_by_EnvFact(hspR,Rv0350))]

Every element in the hypothesis (list) is either a positive or negative ground abducible or a negative ground non-abducible. Starting from the bottom: the gene Rv0350 in experiment hspR is not affected by an environmental factor; the increase in expression of Rv0350 in experiment hspR is not due to the effect of a gene other than the knocked out gene (Rv0353); Rv0352, which is a candidate intermediary gene, does not induce Rv0350 (because there is no experimental or background evidence supporting this hypothesis); and finally, Rv0353 inhibits Rv0350 (since in an experiment where Rv0353 has been knocked out, its expression appears increased and cannot be explained in any other way).

Explaining two observations The second example demonstrates how the recursive rules can identify possible intermediary genes in the interaction network.

The input is *reduces_expression(sigH, Rv2710)*, representing that the expression levels of gene Rv2710 in experiment 'sigH' exceeded the specified significance threshold. 'sigH' is the name of an experiment that knocks out gene Rv3223c/sigH (i.e. it compares a sample with gene Rv3223c/sigH knocked out against the wild type). In this case the system produces two explanatory hypotheses:

Hyp = [induces(Rv3223c,Rv2710)]
Hyp = [induces(Rv3223c,Rv1221), induces(Rv1221,Rv2710)]

(The negative abducibles have been omitted for clarity.)

The first hypothesis states that the experimental observation is explained if Rv3223c induces Rv2710. The second hypothesis identifies a candidate intermediary gene, Rv1221. There is a way of discriminating between these two hypotheses, because there is an experiment, 'sigE', which knocks out Rv1221. Indeed, when we seek to explain the observations *reduces_expression(sigH, Rv2710)* and *reduces_expression(sigE, Rv2710)* together, only one hypothesis is generated:

Hyp = [induces(Rv3223c,Rv1221), induces(Rv1221,Rv2710)]

The ability to analyse many experiments together in this fashion is a distinguishing feature of our method. In general, the more observations we attempt to explain at the same time, the more coherent the hypothesis, as there is more information available to reject alternatives.

Of course, if there is noise or error in the experimental data there may be no hypothesis which can explain all observations. Micro-array data is notoriously unrleiable. The significance thresholds values used in the first-phase statistical analysis of raw data are deliberately chosen to be conservative. We are currently experimenting with the model to determine how sensitive it is to decisions about these significance thresholds.

Heat Shock Response As a final, more realistic, example, selected observations were analysed from five different microarray experiments together. Each knocks out or over-expresses a gene believed to be involved in heat shock response and also known to function as a transcriptional regulator (i.e., a regulator of expression of other genes). The experiments sigE, sigH, hrcA, hspR measure the effects of knocking out genes Rv1221, Rv3223c, Rv2374c, Rv0353, respectively, and experiment dnaJ over-expresses gene Rv0352.

Analysis of the observations in all five experiments together generated a single hypothesis that explained all the observations. The hypothesis is shown in graphical form in figure 1.

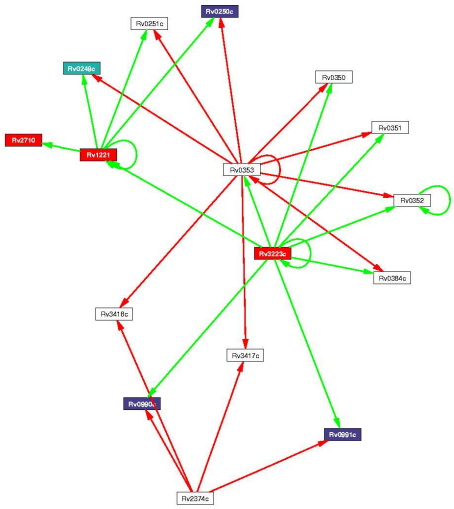

Fig. 1. The nodes represent genes, colour coded according to a standard functional classification. The edges show the inferred relations between genes, red for *inhibits* and green for *induces*. The cyclical edges represent the auto-regulation relationships abduced. In the visualisation tool, operons are drawn as clusters of genes. Clusters are omitted in this diagram to aid readability.

The resulting hypothesis is in agreement with previous knowledge about these regulators [6]. It represents a subset of the heat shock response pathway. For example, the DnaK operon (genes Rv0350–353, appearing on the right of the figure) is controlled by the positive regulator sigH (Rv3223c) and the negative regulator hspR (Rv0353). The acr2 operon (genes Rv0249c–251c, at the top left of the figure) is controlled by the positive regulator sigE (Rv1221) and the negative regulator hspR (Rv0353). The groES/EL genes (Rv3417c–Rv3418c) are under dual negative control by hspR (Rv0353) and hrcA (Rv2374c).

Known feedback loops are also discovered. For example, the DnaK operon (Rv0350–353) is negatively regulated via its own fourth member Rv0353 thus producing a feedback loop.

Finally, there is a group of genes whose function in heat shock response is not clear but which are linked in the explanatory hypothesis generated. For example, the genes Rv0249c and Rv0250c are both unknown genes, both repressed (inhibited) by hspR, both next to Rv0251c (acr2) in the chromosome. This could be a real effect, suggesting that they are in an operon, or it could be some artefact due to their place on the chromosome and the way the microarray chip collects the data. Similarly, Rv0990c and Rv0991c could also be members of an operon, but this time isolated with no obvious function in heat shock. Our biological collaborators are now planning to investigate these hypotheses in a new set of microarray experiments.

4 Application: *S. cerevisiae*

For the purposes of validating our model, the datasets on *M. tuberculosis* are not ideal. The experiments are rather fragmentary, limiting the number of cross-experiment analyses we can conduct, and there is limited knowledge available on regulatory mechanisms in *M. tuberculosis* for validation of the hypotheses.

We are currently applying our methods to datasets on yeast (*Saccharomyces cerevisiae*). This is an extensively studied organism, with large amounts of experimental data available in public databases and extensive annotation of known biological functions. We are employing the datasets [12, 13] to study a larger number of known biological processes and validate/improve the model accordingly.

Yeast is a eukaryotic organism and so more complex than *M. tuberculosis*. Details of the gene interaction model had to be adjusted slightly to comply with the properties of the yeast genome and prior knowledge, including a number of integrity constraints and the definition of regulator genes. However, the general rules of the gene interaction model remained the same.

The work on yeast is still in its initial stages but the results obtained so far are quite promising. For example, one analysis that we have completed involved data from four experiments that knocked out four different transcription regulator genes: YLR442c, YMR280c, YMR047w and YBR083c. The set of observations to be explained consisted of all genes that were affected by more than one experiment. Some of the relations inferred can be given biological support by reference to annotations of the genes in the Comprehensive Yeast Genome Database [13]. For example, the program generated possible relationships between the gene YLR442C and the genes YCL027W and YBR073W. These seem to be meaningful, since gene YLR442C is a regulator of the DNA repair and mating processes and genes YCL027W and YBR073W are genes involved in those processes. Further discussion is omitted because of space limitations.

5 Related Work

The most common approaches to the analysis of microarray data employ various probabilistic methods, including recently the use of Bayesian Networks (see e.g. [2]). These approaches attempt to discover gene regulation relationships directly from raw expression level data, looking for patterns of expression levels

in a single microarray experiment or, in other kinds of microarray experiments not considered in our work, in a series of microarray measurements over time.

To our knowledge, the inference of regulatory networks from microarray data has not previously been formulated as a problem of abduction, though abductive inference has been used in other kinds of genetics experiments. GenePath [8] uses abductive reasoning to construct a genetic network from classical genetics experiments, where instead of differences in expression levels for genes, the experiments evaluate the different phenotypes that occur given gene knock-outs or over-expressions. The nature of the experimental data, of the hypotheses, and consequently of the model itself, are quite different from what is addressed in this paper. In [7] a hybrid framework of ALP and Inductive Logic Programming (ILP) is applied in an attempt to uncover general patterns of interaction in biochemical (metabolic) pathway data. Again, the nature of the input data and the type of hypotheses to be generated are quite different in our work.

More generally, there is increasing interest in modelling biological phenomena using AI methods, focussing in particular on biochemical processes and signalling pathways. There is also a growing body of work on the application of formal tools for modelling concurrent (computer) systems to the analysis of biological networks.

All these latter works, however, are concerned with modelling biological processes at a much finer level of detail than we are addressing in this paper. We are concerned with uncovering possible gene interaction relationships but without yet any information about the actual processes by which these interactions are effected. We are addressing in this paper the analysis of data at a much earlier stage of the scientific process.

6 Evaluation & Future Work

We see the contributions of the work as threefold. First, we are developing a *general method* to support the analysis of an important class of microarray experiments. The novel feature is the use of a simple, general model of how gene interactions can cause changes in observable expression levels of genes under differing conditions, and the use of abduction to infer explanatory hypotheses for the experimental results. This method allows us to infer regulation relations across several experiments.

Second, is the development of the *gene interaction model* itself. We attach particular importance to the declarative and modular nature of this model, which allows us to experiment easily with variations and new general rules suggested by our biological collaborators, and to incorporate relevant biological knowledge as it becomes available. A key feature are the parameteric relations *related_gene* and *intermediary_gene*. These parameters allow our collaborators to focus on particular classes of genes, or genes related by function or biological process. From the computational point of view, they allow us to constrain the search space of possible hypotheses, making it possible to apply the methods in practice.

Third, there are the actual biological results we obtain by applying the methods to the available datasets. These are still at an early stage. The tests per-

formed on *M. tuberculosis* successfully rediscovered part of the heat shock response mechanism and have suggested further microarray experiments to uncover other parts of this mechanism. In yeast, our initial tests have revealed biologically meaningful relations that also suggest improvements to the model. We are presently engaged in a systematic exploration of the various possibilities afforded by the model and an extensive validation of the model against known gene regulation processes in yeast. We do not want to over-state these claims. The experimental results reported here are fragmentary. It is not our aim in this paper to present a detailed evaluation of the biological significance of these results.

It seems to be generally assumed that the ALP methods employed here are too brittle for practical application. We have tried a range of ALP implementations reported in the literature, and it is certainly the case that some of them proved to be insufficiently robust to cope with this application. We are also currently experimenting with alternative, related ways of formulating the problem, as an Answer Set Programming problem for example. Whatever the biological significance of this technique turns out to be in the longer-term, the model provides a valuable test case for those concerned with the development of abductive reasoning technology and related techniques.

References

1. Endriss U., *et al.* The CIFF Proof Procedure for Abductive Logic Programming with Constraints. *JELIA04*, LNAI 3229, Springer-Verlag, pp31–44, 2004.
2. Friedman N., Linial M., Nachman I., Pe'er (2000) Using Bayesian Networks to analyze expression data. *J. Comp. Bio.* 7:601–620.
3. Kakas, A.C., Kowalski, A., Toni, F. (1993) Abductive Logic Programming. *J. of Logic and Computation* 2(6):719–770.
4. Kakas, A.C., Mancarella, P. A simple Abductive Logic Programming (ALP) System. http://www.cs.ucy.ac.cy/aclp/alp_int.pl
5. Papatheodorou I., M. Sergot M., Randall M., Stewart G.R., Robertson B.D. Visualisation of Microarray results to assist interpretation *Tuberculosis* 84:275–281 (2004).
6. Stewart, G. *et al* (2002) Dissection of the heat-shock response in *Mycobacterium Tuberculosis* using mutants and microarrays. *Microbiology* 10:3129–3138.
7. Tamaddoni-Nezhad A., *et al* Modelling inhibition in metabolic pathways through abduction and induction. *ICLP04*, LNAI 3194, Springer-Verlag, pp305–322, 2004.
8. Zupan, B., Demsar, J., Bratko, I. *et al* (2003) GenePath: a system for automated construction of genetic networks from mutant data. *Bioinformatics* 19:383–389.
9. Kakas, A.C., Denecker, M. (2002) Abduction in Logic Programming. In *Computational Logic: Logic Programming and Beyond, Part I.* Springer-Verlag, pp402–436.
10. http://schoolniklab.stanford.edu/projects/tb.htm
11. http://www.sanger.ac.uk/Projects/M_tuberculosis
12. http://www.transcriptome.ens.fr/ymgv/
13. Comprehensive Yeast Genome Database. http://mips.gsf.de/genre/proj/yeast/index.jsp
14. http://www.research.att.com/sw/tools/graphviz