Linköping Electronic Articles in
Computer and Information Science
Vol. 3(1998): nr 7

Developing a Logical Model of Yeast Metabolism

Philip G.K. Reiser

Ross D. King

Douglas B. Kell

Stephen H. Muggleton

Christopher H. Bryant

Stephen G. Oliver

Linköping University Electronic Press
Linköping, Sweden

http://www.ep.liu.se/ea/cis/1998/007/

Published on October 15, 1998 by
Linköping University Electronic Press
581 83 Linköping, Sweden

Linköping Electronic Articles in
Computer and Information Science
ISSN 1401-9841
Series editor: Erik Sandewall

Recommended citation:
<Author>. <Title>. Linköping Electronic Articles in Computer and Information Science, Vol. 3(1998): nr 7.
http://www.ep.liu.se/ea/cis/1998/007/. *October 15, 1998.*

This URL will also contain a link to the author's home page.

Abstract

With the completion of the sequencing of genomes of an increasing number of organisms, the focus of biology is moving to determining the role of these genes (functional genomics). To this end it is useful to view the cell as a biochemical machine: it consumes simple molecules to manufacture more complex ones by chaining together biochemical reactions into long sequences referred to as *metabolic pathways*. Such metabolic pathways are not linear but often intersect to form complex networks. Genes play a fundamental role in these networks by providing the information to synthesise the enzymes that catalyse biochemical reactions. Although developing a complete model of metabolism is of fundamental importance to biology and medicine, the size and complexity of the network has proven beyond the capacity of human reasoning. This paper presents intermediate results in the Robot Scientist research programme that aims to automatically discover the function of genes in the metabolism of the yeast *Saccharomyces cerevisiae*. Results include: (1) the first logical model of metabolism; (2) a method to predict phenotype by deductive inference; and (3) a method to infer reactions and gene function by abductive inference. We describe the *in vivo* experimental set-up which will allow these *in silico* predictions to be automatically tested by a laboratory robot.

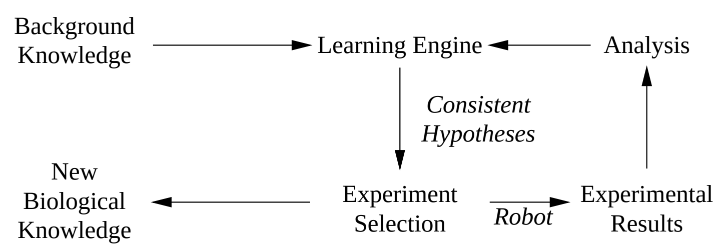

Figure 1: An overview of the Robot Scientist.

1 Introduction

The aim of the *Robot Scientist* project is to construct a physical implementation of a scientific active learning system (see Figure 1). The system will construct an initial set of hypotheses and then repeat the following cycle: (1) devise experiments to select between competing hypotheses, (2) direct a robot to physically perform these experiments, (3) automatically analyse the experimental results, and (4) revise its hypothesis set in the light of the experimental results. This cycle is then repeated until only one hypothesis remains.

The Robot Scientist project is investigating scientific active learning by applying it to the problem of functional genomics to automatically produce useful knowledge about genes of unknown function.

2 Functional Genomics

With the completion of the sequencing of genomes of an increasing number of organisms, the focus of biology is turning towards determining the role of genes. For example, the yeast *S. cerevisiae*, one of the most intensely studied of all organisms, has ≈6,000 predicted protein-encoding genes [6]. Of these, only ≈60% can be assigned a function with any confidence. Most of these genes have had functions proposed based on sequence similarity, but have not been confirmed experimentally. Furthermore, many annotations are incomplete and incorrect as functional assignment by homology is incorrectly assumed to be transitive. The new science of *functional genomics* is dedicated to determining the function of genes of unknown function, and to further detailing the function of genes with purported function.

To meet the challenge posed by functional genomics, new, and highly ingenious experimental techniques have been developed. These techniques permit large-scale, and parallel interrogation of cell states under different stages of development and defined environmental conditions. Such analyses may be carried out at the level of transcription [18, 3] and at the level of translation [21, 11]. Most recently, the metabolome (the cell's small molecule complement) [4, 13], protein interaction assays [20], and large-scale phenotyping [15, 17] have emerged as other important levels for functional genomics studies.

The above experimental techniques are valuable as they provide windows into the internal workings of cells. However, their output is more often a deluge of data than distilled scientific knowledge. To integrate and exploit this experimental data it is essential to develop models of cells that can explain and predict the experimental observations. To this end, we view the cell as a biochemical machine: it consumes simple molecules to manufacture more complex ones by chaining together biochemical reactions into long sequences referred to as *metabolic pathways*. Such metabolic pathways are not linear but often intersect to form complex networks.

2.1 Problems in the Analysis of Metabolism

In the analysis of metabolism important questions are:

Problem 1 *Given a model of metabolism and a set of nutrients, is it possible to synthesise a particular compound?*

Problem 2 *Given a model of metabolism and a particular compound, what nutrients are necessary to synthesise that compound?*

Problem 3 *Given a model of metabolism and a set of nutrients, can a set of defined compounds be synthesised?*

Problem 4 *Given a model of metabolism and a pair of compounds, what is the shortest distance between them (measured in number of reactions or ATP-equivalent)?*

Problem 5 *Given an incomplete/incorrect model of metabolism, a set of nutrients, and a set of compounds that can be synthesised, discover missing/incorrect reactions.*

The size and complexity of metabolic networks has proven beyond the capacity of human reasoning and has hindered our ability to solve the above problems. This is a principal motivation for the use of logic.

3 A Logical Model of Metabolism

In this section, a logical setting is defined for reasoning about metabolism. Reactions and compounds are described as logic formulae in the predicate calculus, and together form a logical model of metabolism. The model's consistency and completeness can be analysed by comparing the model's logical consequences with the outcome of *in vivo* observations in auxotrophic experiments. This permits a clear specification of the problem of inferring missing reactions based on experimental results.

3.1 Metabolic Graphs

Metabolic pathways in an organism are not linear but often intersect to form complex networks. A cell's metabolism is therefore naturally modelled by a graph structure. Metabolism can be represented with (at least) two kinds of graph structure as illustrated in Figure 2.

Classical metabolic graph. Each vertex corresponds to a *single compound*. An edge describes a reaction from one compound to another. This representation shows the key compounds involved in a reaction and is useful because of its clarity.

Complete metabolic graph. Each vertex corresponds to a *set of compounds* that are available to participate in a reaction. There is a unique start vertex corresponding to the set of all nutrients available to the cell. An edge corresponds to a reaction. There can only be an edge when all the reaction's substrates are available in the origin of the edge. The reaction's products are then added to the set of

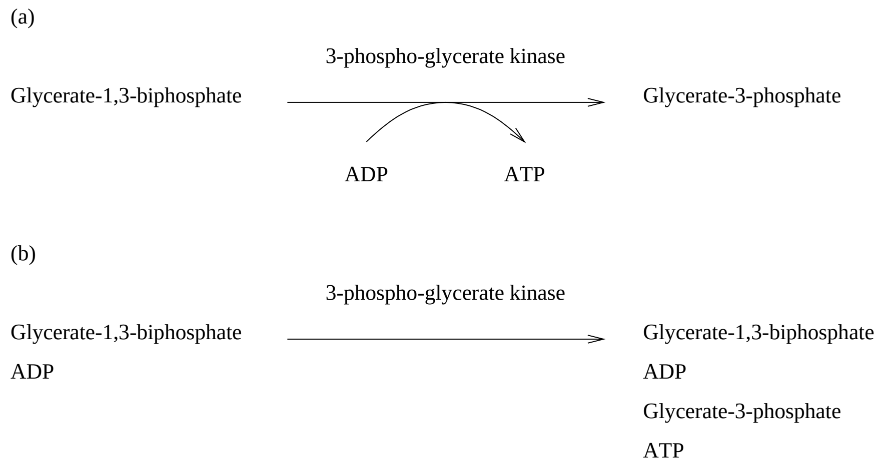

Figure 2: Representation of metabolic graphs. A step in the glycolysis pathway represented (a) with a classical metabolic graph and (b) with a complete metabolic graph.

available compounds at the destination of the edge. As a result, an edge cumulatively increases the compounds available in the cell. The compounds at each vertex is the set of all nutrients and the compounds that can be synthesised by the reactions prior to it. This may increase to include several hundred compounds.

Remark 1 *A reaction is modelled as a unidirectional transformation; the forward and reverse directions of a bidirectional reaction are treated as two separate reactions, i.e. the graph is directed*[1].

The complete metabolic graph is most suited to a computational model of metabolism. As the complete set of available compounds is at hand, the reactions can occur can easily be determined.

3.2 Logical Models of Metabolism

Below we describe a logical setting for metabolism and show how synthesis of compounds corresponds to logical consequence. As a result these compounds can be derived by deduction. Examples are given illustrating both classical and complete metabolic graphs.

The following symbols are used below $\wedge$ (logical and), $\models$ (logically entails) and $\Box$ (falsity). Further details can be found in [8].

Definition 1 (Reactions) *A conjunction of clauses, each corresponding to a reaction*

$$R = r_1 \wedge r_2 \wedge \cdots \wedge r_m.$$

Definition 2 (Pathway) *A relation that composes reactions.*

$$P = p_1 \wedge p_2 \wedge \cdots \wedge p_n.$$

[1] In principle, there are no irreversible reactions in chemistry. Although the equilibrium constant (standard Gibbs energy) will always be finite, it can be immense, e.g. order 10^5 for pyruvate kinase. So, although in theory all enzymes catalyse reversible reactions, it is of no practical consequence.

Definition 3 (Metabolism) *Let R be a set of reactions and P be a relation that composes the reactions.* Logical metabolism *is defined as*

$$M = P \wedge R.$$

Example 1 (Classical metabolic graph) *Let R_1 be defined as the following reactions*

$$R_1 = \left\{ \begin{array}{l} reaction(a, b) \leftarrow \\ reaction(b, c) \leftarrow \end{array} \right.$$

and P_1 be defined as

$$P_1 = \left\{ \begin{array}{l} path(X, Y) \leftarrow reaction(X, Y) \\ path(X, Z) \leftarrow reaction(X, Y) \wedge path(Y, Z) \end{array} \right.$$

Then,

$$M_1 = P_1 \wedge R_1$$

is a logical model of metabolism based on the classical metabolic graph.

Definition 4 (Nutrients) *A conjunction of clauses each corresponding to a compound available to the cell from its growth medium*

$$N = c_1 \wedge c_2 \wedge \cdots \wedge c_p.$$

Definition 5 (Reachable) *Given a model of metabolism M and a set of nutrients N, a compound c is* reachable *if it is a logical consequence of the metabolism and the nutrients*

$$M \wedge N \models c.$$

The reachability of a compound given nutrients and a model of metabolism can therefore be computed using deductive inference. If the above definitions are restricted to logic programs (i.e. conjunctions of definite clauses) then deductive inference can be performed by SLD resolution as implemented in Prolog.

Example 2 (Complete metabolic graph) *Let R_2 be defined as the following reactions*

$$R_2 = \left\{ \begin{array}{l} reaction([a], [a, b]) \leftarrow \\ reaction([b], [b, c]) \leftarrow \end{array} \right.$$

and P_2 be defined as

$$P_2 = \left\{ \begin{array}{l} path(X, Y) \leftarrow reaction(A, B) \wedge subset(A, X) \wedge union(X, B, Y) \\ path(X, Z) \leftarrow reaction(A, B) \wedge subset(A, X) \wedge union(X, B, Y) \wedge path(Y, Z) \end{array} \right.$$

where subset/2 and union/3 are predicates defining conventional set operations. Then,

$$M_2 = P_2 \wedge R_2$$

is a logical model of metabolism based on the complete metabolic graph.

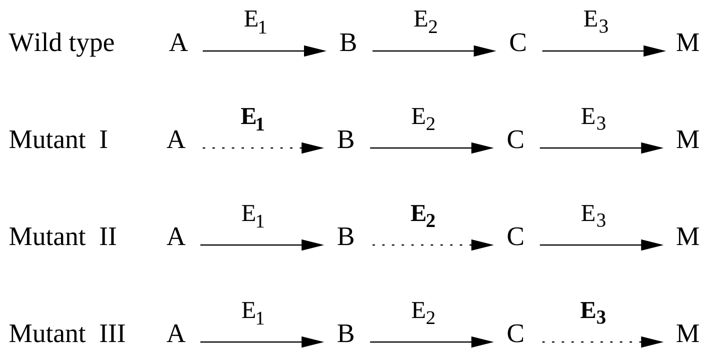

Figure 3: Auxotrophic growth experiments (adapted from [7]). Dotted lines indicate defective pathways.

Strain	Supplement			
	A	B	C	M
Wild type	+	+	+	+
Mutant I	−	+	+	+
Mutant II	−	−	+	+
Mutant III	−	−	−	+

Table 1: Restoring growth of auxotrophs by adding supplements.

3.3 Auxotrophic Growth Experiments

Auxotrophic growth experiments are a classical technique for inferring metabolic pathways in a microorganism [7]. An *auxotrophic mutant* is a strain of an organism that has a mutated or deleted gene so that it is defective in a biosynthetic pathway and as a consequence cannot grow and replicate. However, by adding the normal product of the pathway growth can be restored.

As a concrete example, consider the pathway shown in Figure 3 that synthesises the essential compound M. With the wild type (the strain with no missing genes), M can be synthesised via the precursors A, B and C using the enzymes E_1, E_2 and E_3. However, in mutants I, II and III, gene deletions prohibit one of the enzymes from being synthesised and as a result these mutants will not grow.

Growth can be restored by adding the precursors or the end product. When the compounds A, B, C and M are added to the four strains, the observed growths are shown in Table 1. The growth properties of the single-gene deletion strain in the presence of the appropriate precursors indicates the reaction the enzyme catalyses. Given a gene of unknown function, one can use auxotrophic growth experiments to infer the biochemical role of the enzyme it codes for.

Study of cells has revealed a number of essential molecules that must be present for growth and replication to occur. These molecules include macromolecules such as proteins, nucleic acids (DNA and RNA), phospholipids (cell membrane), lipopolysaccharides and oligosaccharides (cell wall) as well as the following building blocks: amino acids, purines, pyrimidines, lipids, and saccharides. The deletion of any genes essential in the synthesis of these molecules will prevent growth and replication. As a result, auxotrophic experiments can be used to infer their biochemical function.

3.4 Logical Setting for Auxotrophic Growth Experiments

Definition 6 (Logical phenotype) *A* logical phenotype *is an observable property of the organism under study that takes a truth value. A logical phenotype p may be true or false denoted p^+ or p^- respectively.*

In auxotrophic growth experiments, the observable property is growth of the microorganism. The truth values of the logical phenotype correspond to growth and no growth.

Definition 7 (Metabolic baseline) *Let M be a model of metabolism, N be a set of nutrients and p_w be a logical phenotype of the wild type. The* metabolic baseline *for that phenotype is defined as*

$$M \wedge N \models p_w^+$$

Definition 8 (Metabolic necessity) *Let M be a model of metabolism, N be a set of nutrients and p_m be a logical phenotype for mutant strain m.* Metabolic necessity *for that phenotype is defined as*

$$M \wedge N \not\models p_m^+$$

Definition 9 (Metabolic sufficiency) *Let M be a model of metabolism, N be a set of nutrients, p_m be a logical phenotype for mutant strain m, and r be a reaction.* Metabolic sufficiency *for that phenotype is defined as*

$$M \wedge N \wedge r \models p_m^+$$

Definition 10 (Weak consistency) *Let M be a model of metabolism, N be a set of nutrients, r be a reaction. M is said to be* weakly consistent *if*

$$M \wedge N \wedge r \not\models \Box$$

Definition 11 (Strong consistency) *Let M be a model of metabolism, N be a set of nutrients, r be a reaction, and p be a logical phenotype. M is said to be* strongly consistent *if*

$$M \wedge N \wedge r \wedge p^- \not\models \Box$$

Definition 12 (Metabolic completeness)

The logical setting for auxotrophic growth experiments is valuable as it clearly defines the problem to be solved. That is, find one or more reactions to add to the existing model of metabolism so that the model is consistent and complete with respect to the growth observations.

4 Inference of Metabolic Pathways

Inference or reasoning is the derivation of new facts from existing facts or premises by any acceptable form of reasoning. The three main types of logical inference are deduction, induction and abduction [12]. Abduction and induction are related and their distinction is still controversial [5]. One distinction is that abduction infers ground propositions while induction infers non-ground propositions (or rules). According to this distinction, the type of inference most suited to the inferring reactions from auxotrophic growth experiments is abduction.

Abduction is the inference of the case from the general rule and the result. Abduction is often referred to as a form of common-sense reasoning as it can be used to reason from cause to effect.

Example 3 *Given the rule*

$$\textit{rained-last-night} \quad \rightarrow \quad \textit{grass-is-wet}$$

if we observe that the grass is wet, we infer that it rained last night. This is abductive inference and can be seen as the reverse of a deductive reasoning.

4.1 Theory Completion by Abductive Logic Programmming

Definition 13 (Theory Completion) *Let I, T and E be sets of well-formed formulae in the predicate calculus.*

- Given: *integrity constraints I, examples E, and a theory T_{fix} that satisifies I,*
- Find: *a theory $T = T_{fix} \cup T_{add}$ obtained by* **adding clauses** T_{add} *such that: $T \models E$, and T satisfies I.*

There exist several techniques for deriving completions. The approach used by Moyle and Muggleton [9, 1] is *logical backpropagation.* However, logical backpropagation has a number of limitations; most significantly, it cannot infer completions with more than one literal. The implication for inferring metabolic reactions is that it can only infer single reaction steps (see Section 4.3).

Another way of performing theory completion is Yamamoto's Skip Ordered Linear resolution for Definite clauses (SOLD) [22]. SOLD resolution is a variant of SLD resolution refutation [16]. SOLD resolution is a proof technique for deriving a goal clause $\bar{H}$ from a definite program T and a goal $\bar{E}$ where $T \cup \bar{E} \models \bar{H}$.

A theory completion can be found by applying SOLD resolution to the existing theory T_{fix} and the negation of the example $\bar{E}$:

$$T = T_{fix} \cup T_{add}, \quad \text{where } T_{add} \in \text{SOLD}(T_{fix}, \bar{E}).$$

4.2 Graph Inference by Theory Completion I

The use of abductive inference to complete a theory can be illustrated by the following simple application. Consider the graph shown in Figure 4. This graph is defined by the clauses

$$\begin{array}{l} \text{edge}(a, b) \leftarrow \\ \text{edge}(c, d) \leftarrow \end{array}$$

Suppose there is evidence that this graph is incomplete. More specifically, it is known that there exists a path between nodes a and d, i.e.

$$\text{path}(a, d) \leftarrow$$

Figure 4: A simple graph.

where path/2 is defined as

$$\begin{array}{l} \text{path}(X,Y) \leftarrow \text{edge}(X,Y) \\ \text{path}(X,Z) \leftarrow \text{edge}(X,Y), \text{path}(Y,Z) \end{array}$$

Consequently, we have the theory T_{fix} and an example E defined as follows

$$T_{fix} = \begin{cases} \text{edge}(a,b) \leftarrow \\ \text{edge}(c,d) \leftarrow \\ \text{path}(X,Y) \leftarrow \text{edge}(X,Y) \\ \text{path}(X,Z) \leftarrow \text{edge}(X,Y) \wedge \text{path}(Y,Z) \end{cases}$$

$$E = \{ \ \text{path}(a,d) \leftarrow$$

By applying SOLD resolution to the background knowledge and example, the following clauses are inferred. The clauses correspond to sets of edges that could be added to the graph to complete the pathway.

$$\text{SOLD}(T_{fix}, \bar{E}) = \begin{cases} \text{edge}(a,d) \leftarrow \\ \text{edge}(a,c) \leftarrow \\ \text{edge}(a,c) \wedge \text{edge}(d,c) \leftarrow \\ \text{edge}(b,d) \leftarrow \\ \text{edge}(b,c) \leftarrow \\ \text{edge}(b,a) \wedge \text{edge}(b,d) \leftarrow \\ \quad \vdots \end{cases}$$

The completed theory T is constructed by adding one of the completions found by SOLD to the original theory T_{fix}, i.e.

$$T = T_{fix} \cup T_{add}, \quad \text{where } T_{add} \in \text{SOLD}(T_{fix}, E).$$

Clearly, there are many possible completions and integrity constraints can be used to eliminate irrelevant and inappropriate completions. Integrity constraints will be discussed further in Section 5.3.

4.3 Graph Inference by Theory Completion II

It is not always the case that a single edge can complete the graph. SOLD resolution can also be used to infer multiple edges by introducing new vertices. For instance, in the graph inference example above (Section 4.2), the completions inferred by SOLD would also include:

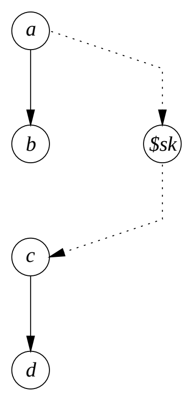

Figure 5: A simple graph with a new vertex and two edges (dotted lines) inferred by SOLD resolution.

$$\mathrm{SOLD}(T_{fix}, \bar{E}) = \begin{cases} \quad \vdots \\ \mathrm{edge}(a, \$sk) \wedge \mathrm{edge}(\$sk, c) \leftarrow \\ \quad \vdots \end{cases}$$

This clause thereby introduces a new vertex into the graph labelled with a Skolem constant $sk. This is illustrated in Figure 5.

Note we are learning the definition of predicates that are unobservable. In the above examples the only observable is the existance of a path in the graph. Yet the predicate being learned is the existence of edges. Abductive logic programming is therefore highly suited to the problem of metabolic reconstruction where the individual reactions are difficult or even impossible to observe, but the existance of a pathway to an essential compound is relatively easy to observe.

5 Developing a Logical Model of Yeast Metabolism

5.1 Data

There has been considerable effort in cataloging current knowledge of biochemistry and molecular biology and making it accessible to the scientific community. The Kyoto Encyclopaedia of Genes and Genomes (KEGG) [**?**] is one example. Most of the data in KEGG consist of information on interacting molecular and gene pathways. Related to KEGG are the Biochemical Pathways (BP) index of Boehringer Mannheim [**?**], and the Encyclopaedia of E. coli Genes and Metabolism (EcoCyc) [**?**].

Almost all existing bioinformatic databases are designed only to store data generated by biological experiments and to make these data available to other biologists. Such databases do not support the kind of reasoning necessary for automatically inferring metabolic pathways. We have therefore developed a logical model of yeast by converting most of KEGG into logical statements. The resulting logical model holds information about genes, the enzymes they code for, the reactions they catalyse and the compounds involved in the reactions. Some details of the model are shown in Table 2.

number of yeast genes	6121
yeast genes of unknown function	5095
yeast genes of known function	1026
number of reactions	5215
number of compounds	5873

Table 2: Properties of the logical model constructed from KEGG.

5.2 Case Study: The Aromatic Amino Acid Biosynthesis Pathway

A logical model of the aromatic amino acid biosynthesis pathways of the yeast *S. cerevisiae* was constructed using the scientific literature (See Appendix B). The model was then compared with the pathways found in KEGG. Reactions were found in KEGG that are do not occur in *S. cerevisiae*. Also there are some reactions missing in KEGG. This illustrates how the even a well-known pathway in a well-studied organism may be inconsistent and complete.

5.3 Metabolic Inference by Theory Completion

The complete model of aromatic amino acid biosynthesis pathways was used as the basis of a rediscovery experiment. When a single reaction was deleted, as occurs with a single-gene deletion, the theory completion approach described in Section 4 infers the set of candidate reactions[2] to restore the defective pathway.

Unconstrained, the set of candidate reactions is necessarily very large and grows exponentially with the number of vertices. However, many of the candidate reactions are chemically unrealistic, for instance a single reaction between the start and end of the pathway may complete the pathway, but cannot be performed by a single enzyme. Integrity constraints are therefore used to limit completions to only those that are chemically plausible using the principle of conservation of mass. This was found to reduce the number of candidate reactions significantly.

5.4 Eliminating Conjectured Reactions by Auxotrophic Growth Experiments

Despite the use of integrity constraints there are typically multiple candidate reactions. The candidate reactions can be viewed as competing hypotheses and by conducting suitable growth experiments we aim to eliminate (falsify) hypotheses until there remains only one.

A single trial on a mutant consists of the selection of a growth medium and the measurement of growth on that medium [15, 17, 19]. Each hypothesis tested will be a prediction of the results of the experiment. Growth is measured by photometry at a single time point after inoculation which gives an easily automated measure of biomass. Errors are controlled by performing replicates of each trial and by using the wild type as a control.

Standard growth experiments are characterised by a very large space of possible growth media. A typical growth medium for *S. cerevisiae* con-

[2]The use of SOLD resolution to construct theory completions allows new reactions to be inferred. In contrast, logical backpropagation [10, 1] cannot infer new reactions but can only select from a complete set of reactions supplied in advance.

sists of carbon sources (at least 50 possible), nitrogen sources (at least 10 possible), other amino acid supplements ($\approx$20 possible), nucleic acid bases and nucleosides (at least 10 possible) and various minerals ($\approx$20 possible). For example a medium could be a combination of 5 carbon sources, 2 nitrogen sources, 10 amino acids, 3 bases and 10 minerals, each component of a medium may be present in varying amounts.

The number of trials that can be performed is clearly prohibitively large and there is a clear need for (1) a high-throughput technique to automate the execution of experiments, and (2) an intelligent way of selecting trials.

6 The Robot Scientist

The aim of the Robot Scientist project is to provide a physical implementation of a scientific active learning system. The study of systems that can choose the next experiment is known as *active learning*. There are two computational tasks in active learning: (1) formation of hypotheses that are consistent with the known background knowledge and experimental results, and (2) selection of the best experiment (or set of experiments) to discriminate between hypotheses[3].

The remainder of this section describes the *Robot Scientist Platform*, the logical interface to the platform referred to as the *Oracle*; and briefly describes the active learning system ASE-Progol.

6.1 The Robot Scientist Platform

The *Robot Scientist Platform* is a robotic system for conducting microbiology assays with minimal human intervention. The platform consists of a laboratory robot, a plate reader and a dedicated PC to control them. The robot is designed to automate the task of liquid handling and can conduct assays by pipetting and mixing liquids on microtitre plates. It is also able to transfer the plates into the adjacent plate reader where measurements can be made using a variety of protocols.

The robot scientist platform may be seen as an oracle that can be queried about the the observable behaviour of micro-organisms. The number of different trials that can be performed is essentially limitless. It is for this express reason that it was built with the objective of connecting it to an intelligent algorithm capable of choosing future experiments based on previous trial outcomes.

6.2 The Hardware

The robot is a Beckman Coulter Biomek 2000 Workstation, a liquid handling robotic workstation[4]. The robot has a work surface consisting of 12 cells. Each cell can hold either microtitre plates, reservoirs for liquids, tips for pipetting, or tools for pipetting and gripping.

The reader is a Wallac Victor2 plate reader[5]. The reader's counting modes cover all the main nonradioactive counting technologies, including

[3] Note that experiment selection in active learning should not be confused with the statistical study of experimental design: the difference is between deciding which question to ask next (active learning) versus ensuring that a set of experiments can answer a question (traditional experimental design).

[4] See http://www.beckman.com for detailed specifications.

[5] See http://lifesciences.perkinelmer.com.

fluorometry, TR-fluorometry, luminometry and photometry. It also has shaking and temperature control features. The server is an IBM PC running Windows NT v3.2. It hosts the robot and reader's software and is connected to the local network and the internet.

6.3 The Oracle

The *Oracle* is the high-level interface to the robot scientist platform. It takes as input a query and returns a response. The query corresponds to an trial and the reponse is the trial's outcome which is established by conducting an assay. Random error can be recognised and eliminated by conducting a number of assay replicates (for details see [14]).

Queries are specified as clauses in the Prolog logic programming language and the results are also returned as Prolog clauses. The Prolog definition of a trial is compiled into a sequence of steps to be performed by the robot and plate reader. These instructions are queued to be executed by the robot. The robot is programmed to execute the experimental procedure shown in Appendix A.

6.4 Closed-Loop Learning

The project will use existing results in active learning theory to develop a method of selecting efficient experiments to discriminate between hypotheses. The system forms an initial set of hypotheses using machine learning, devises experiments to select between competing hypotheses, directs a robot to physical perform these experiments and automatically revises its hypothesis set in the light of the experimental results. This cycle is repeated until it converges on a single hypothesis.

The experimental strategy will seek to minimise the expected cost of finding the best hypothesis. Bryant *et al* [1] are developing an active learning system based on Inductive Logic Programming called ASE-Progol. In this work, they have developed a computationally efficient active learning strategy that is linear in: the number of hypotheses considered, the number of experiments considered, and the time to determine the accuracy of a hypothesis.

7 Discussion

7.1 Graph theoretic versus logical representation of metabolism

Graph theory provides a useful setting for studying a number of properties of metabolism in particular quantitative properties such as metabolic depth and metabolic distance. However, it is less well suited to reasoning about missing reactions, erroneous pathways, etc. Representing metabolism in first order logic allows the benefit of both settings: graphs can be represented naturally in the predicate calculus; in addition, the logical setting facilitates logical reasoning (such as deductive, abductive and inductive inference).

7.2 Limitations of Theory Completion

With theory completion, clauses can be added to the theory, but not removed. As a result, if an incomplete theory includes an incorrect clause, the correct final theory cannot be found by theory completion. Therefore, if the model of metabolism used as the starting point (e.g. taken from KEGG) contains an incorrect reaction, theory completion will not be able to find the correct model of metabolism.

Theory revision [2] is a more general problem setting for discovering logical theories that also allows clauses to be removed. However, it carries the overhead of being computationally significantly more expensive.

7.3 Limitations of Auxotrophic Growth Experiments

Generality of auxotrophic experiments: limited to only those mutant strains that grow on rich media but not on synthetic media. The number of genes that grow on minimal media is yet unknown. Results of the EUROFAN I project examined only a small number of mutant strains. Out of 660 strains, 430 indicated clear growth, 67 did not grow and 163 revealed limited growth.

References

[1] C.H. Bryant, S.H. Muggleton, S.G. Oliver, R.D. King, P.G.K. Reiser, and D.B.Kell. Combining inductive logic programming, active learning and robotics to discover the function of genes. *Machine Intelligence*, 18:??–??, 2001. (to be submitted).

[2] L. De Raedt. *Interactive Theory Revision: An Inductive Logic Programming Approach*. Academic Press, 1992.

[3] J. DeRisi, V. Iyer, and P. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, October 1997.

[4] O. Fiehn, J. Kopka, P. Dormann, T. Altmann, R. N. Trethewey, and L. Willmitzer. Metabolite profiling for plant functional genomics. *Nature Biotechnology*, 18:1157–1161, 2000.

[5] P.A. Flach and A.C. Kakas. *Abduction and Induction: Essays on their relation and integration*. Kluwer Academic Publishers, 2000.

[6] A. Goffeau, Barrell., B., H. Bussey, R. Davis, B. Dujon, H. Feldmann, F. Galibert, J. Hoheisel, C. Jacq, M. Johnston, E. Louis, H. Mewes, Y. Murakami, P. Philippsen, H. Tettelin, and S. Oliver. Life with 6000 genes. *Science*, 274:563–7, 1996.

[7] Albert L. Lehninger. *Biochemistry: the molecular basis of cell structure and function*. Worth Publishers, Inc., 1979. 2nd edition.

[8] J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, Berlin, 1984.

[9] S. Moyle and S. Muggleton. Learning programs in the event calculus. In S. Džeroski and N. Lavrač, editors, *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1297 of *Lecture Notes in Artificial Intelligence*, pages 205–212. Springer-Verlag, 1997.

[10] Stephen Muggleton and Christopher Bryant. Theory completion using inverse entailment. In J. Cussens and A. Frisch, editors, *Proceedings of the 10th International Conference on Inductive Logic Programming*, volume 1866 of *Lecture Notes in Artificial Intelligence*, pages 130–146. Springer-Verlag, 2000.

[11] S.G. Oliver. Proteomics: guilt-by-association goes global. *Nature*, 403:601–603, 2000.

[12] C.S. Pierce. Collected papers of Charles Sanders Pierce. Vol.2, 1931-1958.

[13] L. M. Raamsdonk, B. Teusink, D. Broadhurst, N. Zhang, A. Hayes, M. C. Walsh, J. A. Berden, K. M. Brindle, D. B. Kell, J. J. Rowland, H. V. Westerhoff, K. van Dam, and S. G. Oliver. A functional genomics strategy that uses metabolome data to reveal the phenotype of silent mutations. *Nature Biotech*, pages 45–50, 2001.

[14] P.G.K. Reiser and R.D. King. The robot scientist platform. Internal project report, 2001.

[15] K.-J. Rieger and G. Orlowska et al. Large-scale phenotypic analysis in microtitre plates of mutants with deleted opern reading frames from yeast chromosome iii: Key-step between genomic sequencing and protein function. *Methods in Microbiology*, 28:205–227, 1999.

[16] J. A. Robinson. A machine-oriented logic based on the resolution principle. *JACM*, 12(1):23–41, January 1965.

[17] P. Ross-Macdonald et al. Large-scale analysis of the yeast genome by transposon tagging and gene disruption. *Nature*, 402:413–418, Nov 1999.

[18] M. Schena, D. Shalon, R. Heller, A. Chai, P. O. Brown, and R. W. Davis. Parallel human genome analysis - microarray-based expression monitoring of 1000 genes. *Proc. Natl. Acad. Sci.*, 93:10614–10619, 1996.

[19] F. Sherman. Getting started with yeast. *Methods in Enzymology*, 194:3–21, 1991.

[20] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. QureshiEmili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. J. Yang, M. Johnston, S. Fields, and J. M. Rothberg. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403:623–627, 2000.

[21] M. R. Wilkins, K. L. Williams, R. D. Appel, and D. F. Hochstrasser. *Proteome research: new frontiers in functional genomics.* Springer, Berlin, 1997.

[22] A. Yamamoto. Representing inductive inference with SOLD-resolution. In P. Flach and A. Kakas, editors, *Proceedings of the IJCAI workshop on Abduction and Induction in AI*, pages 59–63, 1997.

Acknowledgements

The authors would like to thank Steve Moyle for his generous help with abductive logic programming and Nathalie Santre for fruitful discussions on auxotrophic growth experiments.

A Experimental Procedure

The experimental procedure executed by the robot scientist platform:

1. for each microtitre plate:
   (a) prepare growth media in all wells.
   (b) add inoculum.
   (c) mix all wells.
2. allow to incubate.
3. for each microtitre plate:
   (a) mix all wells.
   (b) transfer a small sample from each well to a measurement plate.
   (c) move the flat plate to the plate reader.
   (d) measure optical density.

B A Logical Model of Aromatic Amino Acid Biosynthesis in *S. cerevisiae*

The logical model (represented in Progol) is shown in Figures 6 and 7. In Figure 6, the predicate start/1 lists the compounds (using the KEGG labelling scheme) that are assumed to be available to the cell. The predicate end/1 lists the essential compounds that must be synthesised for growth to be observed.

In Figure 7, the predicate enzyme/4 defines the reactions in the pathway. The first argument is a list of ORF names (putative genes); the second argument is the enzyme labelled by Enzyme Commission (EC) number[6]. The third and fourth arguments are lists of substrates and products in the reaction catalysed by the enzyme.

```
start('C00001').
start('C00011').
start('C00074').
start('C00279').
start('C00014').
start('C00064').
start('C00025').
start('C00005').
start('C00006').
start('C00002').
start('C00009').
start('C00065').
start('C00119').
start('C00026'). % needed for reverse reactions
start('C00661'). % needed for reverse reactions

% the following required to get from Tryp to Tyr & Phen
start('C00013').
start('C00022').

% start('edta'). % example inhibitor


end('C00078').
end('C00079').
end('C00082').
```

Figure 6: A Prolog model of Aromatic Amino Acid Synthesis. Part I: start and end compounds.

[6]Enzyme Nomenclature, Nomenclature committee of the International Union of Biochemistry and Molecular Biology. http://www.chem.qmw.ac.uk/iubmb/enzyme/.

```
enzyme(['YBR249C'],'4.1.2.15',['C04691','C00009'],['C00001','C00074','C00279']).

enzyme(['YDR035W'],'4.1.2.15',['C04691','C00009'],['C00001','C00074','C00279']).

enzyme(['YDR127W'],'4.6.1.3',['C04691'],['C00944','C00009']).
enzyme(['YDR127W'],'4.2.1.10',['C00944'],['C00001','C02637']).
enzyme(['YDR127W'],'X',['C02652'],['C02637']). % Isomers
enzyme(['YDR127W'],'1.1.1.25',['C00493','C00006'],['C02652','C00005']).
enzyme(['YDR127W'],'2.7.1.71',['C00493','C00002'],['C03175','C00008']).
enzyme(['YDR127W'],'2.5.1.19',['C00074','C03175'],['C00009','C01269']).

enzyme(['YGR254W'],'4.2.1.11',['C00631'],['C00001','C00074']).

enzyme(['YHR174W'],'4.2.1.11',['C00631'],['C00001','C00074']).

enzyme(['YMR323W'],'4.2.1.11',['C00631'],['C00001','C00074']).

enzyme(['YGL148W'],'4.6.1.4',['C01269'],['C00251','C00009']).

enzyme(['YER090W'],'4.1.3.27',['C00251','C00014'],['C00001','C00108','C00022']).

enzyme(['YER090W','YKL211C'],'4.1.3.27',['C00251','C00064'],['C00108','C00022','C00025']). % note both enzymes needed for activity

enzyme(['YKL211C'],'4.1.1.48',['C01302'],['C00001','C00011','C03506']).  % s/SDS auxotrophic mutant

enzyme(['YDR354W'],'2.4.2.18',['C04302','C00013'],['C00108','C00119']).

enzyme(['YDR007W'],'5.3.1.24',['C04302'],['C01302']).

enzyme(['YGL026C'],'4.2.1.20',['C00065','C03506'],['C00001','C00078','C00661']).
enzyme(['YGL026C'],'4.2.1.20',['C00065','C00463'],['C00001','C00078']).
enzyme(['YGL026C'],'4.2.1.20',['C03506'],['C00463','C00661']).

enzyme(['YPR060C'],'5.4.99.5',['C00251'],['C00254']).

enzyme(['YNL316C'],'4.2.1.51',['C00254'],['C00001','C00011','C00166']).

enzyme(['YGL202W'],'2.6.1.7',['C00166','C00025'],['C00079','C00026']). %not in KEGG
enzyme(['YGL202W'],'2.6.1.7',['C01179','C00025'],['C00082','C00026']). %not in KEGG

enzyme(['YHR137W'],'2.6.1.7',['C00166','C00025'],['C00079','C00026']). %not in KEGG
enzyme(['YHR137W'],'2.6.1.7',['C01179','C00025'],['C00082','C00026']). %not in KEGG

enzyme(['YBR166C'],'1.3.1.13',['C00254','C00006'],['C01179','C00005']).
```

Figure 7: A Prolog model of Aromatic Amino Acid Synthesis. Part II: enzymatic reactions.