Linköping Electronic Articles in
Computer and Information Science
Vol. 6(2001): nr 024

# 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/2001/024/

*Published on November, 2001 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. 6(2001): nr 024.*
http://www.ep.liu.se/ea/cis/2001/024/. *November, 2001.*

*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 the first results of 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.

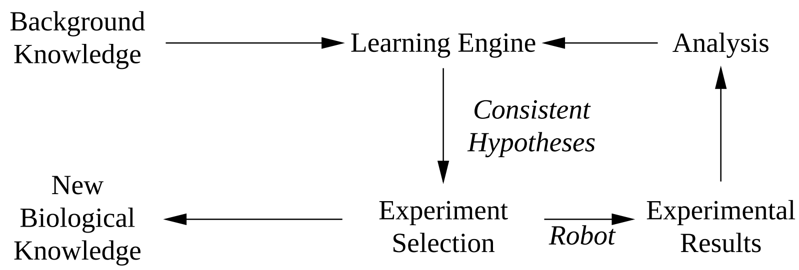


Figure 1: An overview of the Robot Scientist.

# 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 [9]. 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 [35, 6] and at the level of translation [40, 26]. Most recently, the metabolome (the cell's small molecule complement) [7, 28], protein interaction assays [39], and large-scale phenotyping [31, 34] 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 each of a defined set of compounds?*

**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 pair of compounds, what is the shortest distance between them (measured in number of reactions or energy-equivalents(ATP))?*

**Problem 4** *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 experimental results and thereby permits a clear specification of the problem of inferring missing reactions based on *in vivo* observations.

### 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. The classical metabolic graph is reviewed and a new type of metabolic graph, the *complete metabolic graph* is presented.

A reaction is a pair $(l, r)$ where $l$ and $r$ are sets of compounds corresponding to the substrates and products respectively. 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].

**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.

[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, they are not assumed to be reversible in this model.

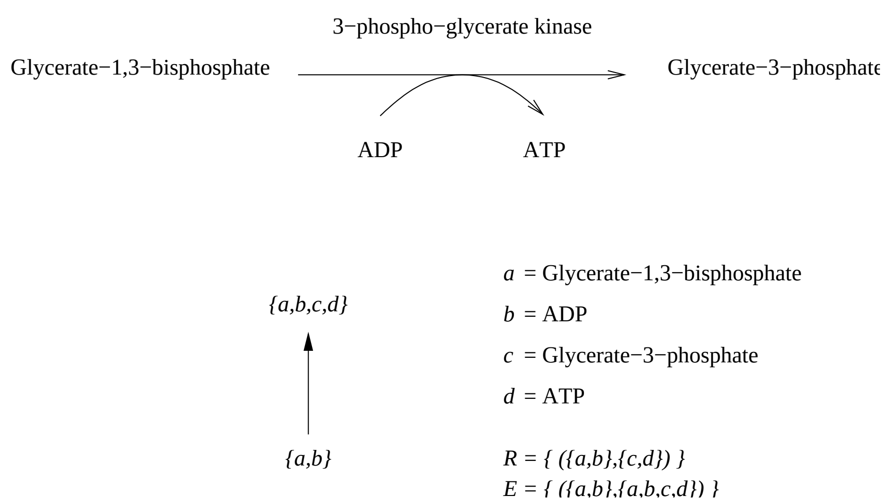


Figure 2: Representation of metabolic graphs. A single step in the glycolysis pathway as typically represented in the literature and represented as a complete metabolic graph.

**Complete metabolic graph.** Each vertex corresponds to a *set of compounds* that are available in the cell. These compounds may be used directly to form parts of the cell or to partipate in further reactions. The graph has a unique start vertex corresponding to the nutrients available to the cell. An edge corresponds to a reaction and the destination of an edge is the set of available compounds plus the reaction's products. As a result, a pathway corresponds to a monotonically increasing set of compounds available in the cell. Some vertices correspond to large numbers (several hundred) of compounds. See Figure 2.

Let the nutrients available to a cell $N$ be a set of compounds.

**Definition 1 (Complete metabolic graph)** *Let $R$ be a set of reactions $R = \{(l_1, r_1), (l_2, r_2), \ldots\}$ and $N$ be a set of nutrients. A* complete metabolic graph *is a directed acyclic graph $G = (V, E)$ where vertices are defined as the union of a number of vertex sets $V = V_0 \cup V_1 \cup \cdots$ and vertex sets $V_i$ are defined recursively as follows*

$$\begin{aligned} V_0 &= N \\ V_{n+1} &= \{v^{'} \mid (l, r) \in R,\ v \in V_n,\ l \subseteq v,\ v^{'} = v \cup r,\ v \subset v^{'}\} \end{aligned}$$

*and edges $E$ are defined*

$$E = \{(u, v) \mid u \in V_n,\ v \in V_{n+1},\ u \subset v\}.$$

The complete metabolic graph has a minimal element (bottom) and a maximal element (top). The bottom corresponds to the cell's nutrients and the top corresponds to all the compounds that are available to the cell that have been either provided as a nutrient or that can be synthesised. For any path $v_1, v_2, \ldots, v_n$ where $(v_i, v_{i+1}) \in E$, the following holds $v_1 \subset v_2 \cdots \subset v_n$, i.e. the set of compounds available to the cell increases monotonically with every biochemical reaction.

**Example 1** *Let the set of nutrients be $N = \{a\}$ and reactions*

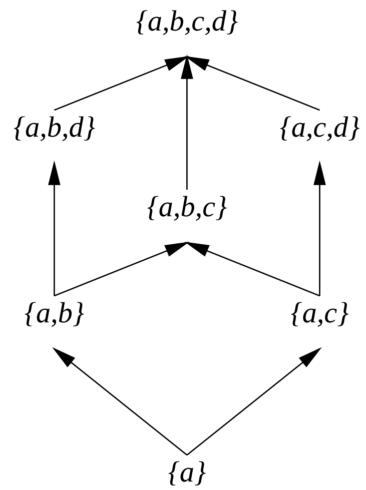


Figure 3: Representation of metabolic graphs. A single step in the glycolysis pathway as typically represented in the literature and represented as a complete metabolic graph.

$$R = \left\{ \begin{array}{c} (\{a\},\{b\}) \\ (\{a\},\{c\}) \\ (\{b\},\{d\}) \\ (\{c\},\{d\}) \end{array} \right\}.$$

*From this the vertex sets $V_i$ are computed:*

$$\begin{array}{ll} V_0 & = \{\{a\}\} \\ V_1 & = \{\{a,b\},\{a,c\}\} \\ V_2 & = \{\{a,b,c\},\{a,b,d\},\{a,c,d\}\} \\ V_3 & = \{\{a,b,c,d\}\} \\ V_4 & = \emptyset \end{array}$$

*and the vertices given by* $V = \{\{a\}\} \cup \{\{a,b\},\{a,c\}\} \cup \cdots$. *Edges are computed from the vertex sets*

$$E = \left\{ \begin{array}{l} (\{a\},\{a,b\}) \\ (\{a\},\{a,c\}) \\ (\{a,b\},\{a,b,c\}) \\ (\{a,b,c\},\{a,b,c,d\}) \\ \vdots \end{array} \right\}.$$

$G = (V,E)$ *is complete metabolic graph and is shown in Figure 3.*

Edges of the complete metabolic graph are labelled with the enzymes that catalyse the reactions and the genes that code for these enzymes. Edges are labelled with the pair $(g,e)$, where $g$ is the gene name and $e$ denotes the enzyme name.

*The complete metabolic graph is better suited than the classical graph to the problem of relating predictions to* in vivo *observations and reasoning about missing reactions.*

### 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.

The following symbols are used below $\wedge$ (logical and), $\models$ (logically entails) and $\Box$ (falsity). Further details can be found in [17].

**Definition 2 (Metabolism)** *Let $G = (V, E)$ be a complete metabolic graph. A logical model of* metabolism *is a conjunction of clauses that represents exactly the edges in $E$ and the paths constructed from these edges.*

**Example 2** *Let $G = (V, E)$ be the complete metabolic graph defined in Example 1. A logical model of metabolism $M$ is defined intensionally with reactions $R$ and predicates for deriving the edges and paths*

$$M = \begin{cases} \mathit{reaction}(\{a\},\{b\}) \leftarrow \\ \mathit{reaction}(\{a\},\{c\}) \leftarrow \\ \mathit{reaction}(\{b\},\{d\}) \leftarrow \\ \mathit{reaction}(\{c\},\{d\}) \leftarrow \\ \mathit{edge}(X,Y) \leftarrow \mathit{reaction}(A,B) \wedge \mathit{subset}(A,X) \wedge \mathit{union}(X,B,Y) \\ \mathit{path}(X,Y) \leftarrow \mathit{edge}(X,Y) \\ \mathit{path}(X,Z) \leftarrow \mathit{edge}(X,Y) \wedge \mathit{path}(Y,Z) \end{cases}$$

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

**Definition 3 (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 4 (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*

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

Therefore it is possible to compute whether a particular compound can be reached, given a set of nutrients and a model of metabolism, by 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.

### 3.3 Auxotrophic Growth Experiments

*Auxotrophic growth experiments* are a classical technique for inferring metabolic pathways in a microorganism [16]. 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 4 that synthesises the essential compound $S$. With the wild type (the strain with

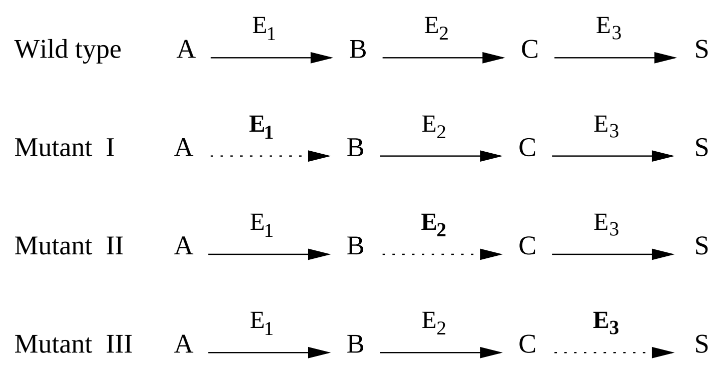


Figure 4: Auxotrophic growth experiments (adapted from [16]). Dotted lines indicate defective pathways.

| Strain | Supplement | | | |
|---|---|---|---|---|
| | A | B | C | S |
| Wild type | + | + | + | + |
| Mutant I | − | + | + | + |
| Mutant II | − | − | + | + |
| Mutant III | − | − | − | + |

Table 1: Restoring growth of auxotrophs by adding supplements.

no missing genes), $S$ 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 $S$ 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 that the enzyme catalyses. Given a gene of unknown function, one can use auxotrophic growth experiments to infer the biochemical role of the enzyme that 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 the biochemical function of genes.

## 3.4 Logical Setting for Auxotrophic Growth Experiments

**Definition 5 (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 6 (Metabolic baseline)** *Let $M_w$ be a model of the metabolism of the wild type, $N$ be a set of nutrients and $p$ be a logical phenotype. The* metabolic baseline *for that phenotype is defined as*

$$M_w \wedge N \models p^+$$

**Definition 7 (Metabolic necessity)** *Let $M_m$ be a model of the metabolism of a mutant strain $m$, $N$ be a set of nutrients and $p$ be a logical phenotype.* Metabolic necessity *for that phenotype is defined as*

$$M_m \wedge N \not\models p^+$$

**Definition 8 (Metabolic sufficiency)** *Let $M_m$ be a model of the metabolism for mutant strain $m$, $N$ be a set of nutrients, $p$ be a logical phenotype, and $r$ be a reaction.* Metabolic sufficiency *for that phenotype is defined as*

$$M_m \wedge N \wedge r \models p^+$$

**Definition 9 (Weak consistency)** *Let $M_m$ be a model of the metabolism of a mutant strain $m$, $N$ be a set of nutrients, $r$ be a reaction. $M_m$ is said to be* weakly consistent *if*

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

**Definition 10 (Strong consistency)** *Let $M_m$ be a model of the metabolism of mutant strain $m$, $N$ be a set of nutrients, $r$ be a reaction, and $p$ be a logical phenotype. $M_m$ is said to be* strongly consistent *if*

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

The problem of inferring reactions and gene function with auxotrophic growth experiments can be defined as:

**Definition 11 (Pathway Discovery)**

- Given: *a model $M_m$ of the metabolism of a mutant strain $m$ and a logical phenotype $p$ that satisfy* metabolic baseline *and* necessity *conditions*
- Find: *one (or more) reaction(s) $r$ such that the new model of metabolism $M_w = M_m \wedge r$ satifies* sufficiency *and* strong consistency *conditions.*

## 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. Three main types of logical inference are deduction, induction and abduction [27]. Abduction and induction are related and their distinction is still controversial [8]. 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 deductive reasoning.*

## 4.1 Theory Completion by Abductive Logic Programmming

**Definition 12 (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 [23]. The approach used by Moyle and Muggleton [24, 4] is *logical backpropagation*. However, logical backpropagation has a number of limitations; most significantly, it cannot infer clauses with more than one literal. The implication for inferring metabolic reactions is that it can only infer single reaction steps. As there are cases where an enzyme catalyses several reactions, logical propagation cannot discover these reactions.

Another way of deriving completions is Yamamoto's Skip Ordered Linear resolution for Definite clauses (SOLD) [41]. SOLD resolution is a variant of SLD resolution refutation [32]. 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 $\bar{E}$:

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

## 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 5. This graph is defined by the clauses

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

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.

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

where path/2 is defined as

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

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

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

Figure 5: A simple graph.

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

By applying SOLD resolution to the background knowledge and example, the following set of clauses is inferred

$$\text{SOLDR}(T_{fix}, \bar{E}) = \left\{ \begin{array}{l} \text{edge}(a, d) \\ \text{edge}(a, c) \\ \text{edge}(a, c) \wedge \text{edge}(d, c) \\ \text{edge}(b, d) \\ \text{edge}(b, c) \\ \text{edge}(b, a) \wedge \text{edge}(b, d) \\ \quad \vdots \end{array} \right\}$$

The clauses correspond to sets of edges that could be added to the graph to complete the pathway. The completed theory $T$ is constructed by adding one of the completions found by SOLD to the original theory $T_{fix}$, i.e.

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

## 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:

$$\text{SOLDR}(T_{fix}, \bar{E}) = \left\{ \begin{array}{l} \quad \vdots \\ \text{edge}(a, \$sk) \wedge \text{edge}(\$sk, c) \\ \quad \vdots \end{array} \right\}$$

This clause thereby introduces a new vertex into the graph labelled with a Skolem constant \$sk. This is illustrated in Figure 6.

In a graph with $v$ vertices, there may be up to $n = v(v-1)$ edges. For those cases where only a single edge is required to complete the graph, the number of possible completions is therefore $O(v^2)$. More generally, if a completion requires up to $r$ edges the number of completions is given by

$$\sum_{i=1}^{r} {}^{n}C_{i}.$$

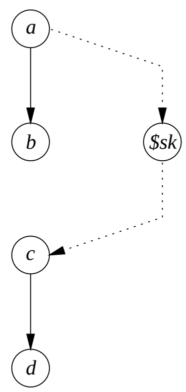


Figure 6: A simple graph with a new vertex and two edges (dotted lines) inferred by SOLD resolution.

For completions with only a small number of edges ($r \ll n$) the number of completions is $O(v^r)$, i.e. the number of completions is polynomial in the number of vertices and exponential in the number of edges allowed in a completion.

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

Note we are learning the definitions 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) [11] is one example. Most of the data in KEGG consists of information on interacting molecular and gene pathways. Related to KEGG are the Biochemical Pathways (BP) index of Boehringer Mannheim [21], and the Encyclopaedia of E. coli Genes and Metabolism (EcoCyc) [12].

Almost all existing bioinformatic databases are designed only to store data generated by biological experiments and to make these data available to the scientific community. 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 assigned function | 1026 |
| yeast genes of unassigned function | 5095 |
| 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 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 even a well-known pathway in a well-studied organism may be inconsistent and incomplete.

### 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 can grow rapidly. The number of reactions is polynomial in the number of vertices, and exponential in the number of reactions that may be added. 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 [31, 34, 37]. 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 [25, 4] 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 (≈20 possible), nucleic acid bases and nucleosides (at least 10 possible) and various minerals (≈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 NT4. 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 a 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.

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 physically perform these experiments and automatically revises its hypothesis set in the light of the experimental results. This cycle is repeated until the robot scientist converges on a single hypothesis.

The experimental strategy will seek to minimise the expected cost of finding the best hypothesis. Bryant *et al* [4] 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 Related Work

Metabolism has been modelled with a variety of representations such as differential equations [18, 19, 20], boolean networks [1], petri-nets [29, 10], $\pi$-calculus [30], graphs [14, 13] and propositional logic [3]. However, no representation is optimal as the choice depends on the problem that is to be solved. For instance, differential equations are well suited to representing the concentrations of metabolites along a pathway and identifying rate-limiting steps but ill-suited to identifying pathways because there are no existing techniques for inferring models with missing reactions. A representation very similar to the complete metabolic graph was developed independently by Karp and co-workers [33]. The Karp approach lacks a rigorous definition and it is not clear how it would be used to infer reactions.

First order logic allows the natural representation of networks and has formalised methods for reasoning over these structures (such as deductive, abductive and inductive inference). This makes first order logic well suited to reasoning about missing reactions and erroneous pathways.

The general problem of reverse engineering pathways can be stated as follows: given a set of input-output pairs of observations of the cell, derive biochemically plausible models of the pathways that can account for these observations. Existing approaches can be classified along three dimensions: (1) the type of observation, (2) the pathway representation and (3) the pathway inference method. Previously used observations include gene expression time-series [1] and metabolite concentration time-series [2, 15]. Types of pathway representation have already been discussed above. Pathway inference methods include: systematic enumeration [14, 36, 22] and genetic programming [15].

# 8 Discussion

## 8.1 Limitations of Auxotrophic Growth Experiments

Auxotrophic growth experiments cannot be used to infer function for all genes. The technique is applicable to only those mutant strains that grow on rich media but not on minimal synthetic media. At present it is not known exactly how many yeast mutant strains grow on minimal media. However, the EUROFAN I project has examined 660 strains of which 67-163 are suitable for auxotrophic growth experiments. In the remaining cases, there are several genes that code for an enzyme or there are redundant pathways and as a result there is no clear phenotypic effect from knocking out a single gene. These cases could be addressed by multiple gene deletions or by adding compounds known to inhibit specific enzymes.

## 8.2 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 [5] 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 more expensive than theory completion.

# References

[1] T. Akutsu, S. Miyano, and S. Kuhara. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, 16(8):727–734, 2000.

[2] A. Arkin, P.D. Shen, and J.Ross. A test case of correlation metric construction of a reaction pathway from measurements. *Science*, 277:1275–1279, 1997.

[3] D.L. Brutlag, A.R. Galper, and D.H. Millis. Knowledge-based simulation of dna metabolism: prediction of enzyme action. *Comput. Appl. Biosci.*, 7(1):9–19, 1991.

[4] 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. (submitted).

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

[6] 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.

[7] 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.

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

[9] A. Goffeau, B. Barrell, 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.

[10] Peter J.E. Goss and Jean Peccoud. Quantitative modeling of stochastic systems in molecular biology by using stochastic petri nets. *Proc. Natl. Acad. Sci.*, 95:6750–655, 1998.

[11] S. Goto, T. Nishioka, and M. Kanehisa. LIGAND: chemical database of reactions. *Nucleic Acids Research*, 28(1):380–382, 2000.

[12] P.D. Karp, M. Riley, S.M. Paley, et al. Eco cyc: Encyclopedia of *escherichia coli* genes and metabolism. *Nucleic Acids Research*, 27(1):55–58, 1999.

[13] M.C. Kohn and D.R. Lemieux. Identification of regulatory properties of metabolic networks by graph theoretical modeling. *Journal of Theoretical Biology*, 150(1):3–25, 1991.

[14] M.C. Kohn and W.J.Letzkus. A graph-theoretical analysis of metabolic regulation. *Journal of Theoretical Biology*, 100(2):293–304, 1983.

[15] J.R. Koza, w.Mydlowec, G.Lanza, J.Yu, and M.A.Keane. Reverse engineering of metabolic pathways from observed data using genetic programming. In *Proc. Pac. Symp. Biocomput.*, pages 434–445, 2001.

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

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

[18] P. Mendes. GEPASI: a software for modeling the dynamics, steady states and control of biochemical and other systems. *Comput. Appl. Biosci.*, 9(5):563–571, 1993.

[19] P. Mendes and D.B. Kell. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14:869–883, 1998.

[20] P. Mendes and D.B. Kell. MEG (Model Extender for Gepasi): a program for the modelling of complex, heterogeneous cellular systems. *Bioinformatics*, 17:288–289, 2001.

[21] G. Michal. *Biochemical Pathways: an atlas of biochemistry and molecular biology.* Wiley, Heidelberg, 1999.

[22] J.E. Mittenthal, B. Clarke, T.G. Waddell, and G.Fawcett. A new method for assembling metabolic networks, with application to the krebs citric acid cycle. *Journal of Theoretical Biology*, 208(3):361–382, 2001.

[23] Stephen Moyle. *An investigation into theory completion techniques in Inductive Logic Programming.* PhD thesis, University of Oxford, 2001. (submitted).

[24] 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.

[25] 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.

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

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

[28] 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.

[29] V.N. Reddy, M.L. Mavrovouniotis, and M.N. Liebman. Petri-net representations in metabolic pathways. In *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, volume 1, pages 328–336, 1993.

[30] A. Regev, W. Silverman, and E.Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. In *Proc. Pac. Symp. Biocomput.*, pages 459–470, 2001.

[31] K.-J. Rieger 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.

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

[33] P.R. Romero and P.Karp. Nutrient-related analysis of pathway/genome databases. In *Proc. Pac. Symp. Biocomput.*, pages 471–482, 2001.

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

[35] 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.

[36] H.Seo, D.Y. Park, L.T. Fan, S. Shafie, B.Bertok, and F.Friedler. Graph-theoretical identification of pathways for biochemical reactions. *Biotechnology Letters*, 23(19):1551–1557, 2001.

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

[38] T.E.Ideker, V.Thorsson, and R.M.Karp. Discovery of regulatory interactions through perturbation: inference and experimental design. In *Proc. Pac. Symp. Biocomput.*, pages 305–316, 2001.

[39] 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.

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

[41] 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, Horst Holstein, Nathalie Santre and Ashwin Srinivasan for fruitful discussions and suggestions.

# 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 7 and 8. In Figure 7, 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 8, 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 7: 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 8: A Prolog model of Aromatic Amino Acid Synthesis. Part II: enzymatic reactions.