MANAGING AND QUERYING MAMMALIAN METABOLIC NETWORKS: A METABOLISM QUERY LANGUAGE AND ITS QUERY PROCESSING

Ali Cakmak[1*], Gultekin Ozsoyoglu[1], and Richard W. Hanson[2]

[1]*Departmen of Electrical Eng. and Computer Science, Case Western Reserve University*
[2]*Department of Biochemistry, Case Western Reserve University*
Cleveland, OH 44106, USA
[*]*cakmak@case.edu*

A metabolic network describes how different cellular processes (i.e., pathways) are connected and work together as part of a metabolism. The complexity of metabolisms and their working principles increase dramatically in higher order organisms (e.g., mammalians) in comparison to relatively lower level organisms (e.g., prokaryotes). As the complexity increases, managing, querying, and visualizing metabolic network data computationally in a manner that is faithful to the underlying biochemistry becomes more and more challenging. In this paper, we present a metabolism query language (MQL). MQL allows the formulation of a wide variety of in-depth metabolism queries, and its query processing heavily employs metabolic principles.

1. INTRODUCTION

Metabolism of an organism involves biochemical processes that perform essential cellular actions, such as biosynthesis and degradation of macromolecules, supplying energy needs of the cell, and so on. Computationally capturing and querying metabolism data is useful both in systems biology research as well as a teaching tool. In the literature, there are mainly two classes of works which study modeling and querying of metabolism. The first class includes query languages (e.g., PQL [1], bcnQL [2]) on biochemical networks, and the second class includes well-known metabolic data sources (e.g., KEGG [3], PathCase [4], MetaCyc [5], Reactome [6]) which house a number of metabolism querying tools built on top of a biochemical network knowledgebase. Such query languages and metabolic data sources essentially view the metabolism as a graph, and mostly focus on querying (i) structural properties of metabolic networks (e.g., paths, neighborhoods, cycles, etc.), and (ii) entity relationships (e.g., inhibitors of a reaction in a particular pathway). However, these efforts do not capture detailed biochemical working principles of a metabolism, and interrelationships between pathways under different physiological and dietary states. We compare the related work briefly in Section 2.

In this paper, we propose a a metabolism query language (*MQL*) that considers (i) the behavior of the metabolic network under different conditions, (ii) metabolic specialization of tissues and subcompartments, (iii) interplay between different pathways, and (iv) distinct pools of metabolites. *MQL* allows for the specification of different classes of queries, such as (i) exploring activated/inactivated paths under specified metabolic conditions, (ii) searching for potential futile cycles, (iii) querying for regulatory changes that prevent a particular futile cycle, (iv) searching for conditions which lead to the (in)activation of a user-specified metabolic subnetwork, and (v) exploring the behavior of a set of (possibly reversible) reactions. And, MQL lets users to input concentration change statements on key metabolites, and incorporates such input into its query processing.

To demonstrate the capabilities of this framework, we employ as an example computational modeling of mammalian (particularly human) metabolism, and specifying and processing queries over its metabolic network. In this paper, we focus on the class of queries regarding *exploring activated/ inactivated paths under specified metabolic conditions*, which we refer to as MQL_{AIP} queries. Please see [8] for the remaining types of queries that can be specified in *MQL*. Next, we informally specify MQL_{AIP} queries via a "template", and illustrate with an example.

1.1. A Query Template and Its Instance

Given:

I. A subset P of pathways in the human metabolic network

II. A set of biological compartments

III. A set C of conditions specifying metabolic and dietary states/physiological conditions, such as fasting, exercise, or specific disease states like diabetes, **and/or** concentration changes (increases/ decreases) of "key metabolites" such as increases in lactate, pyruvate, or amino acids.

(a) Find activated (increased flux) and inactivated (decreased flux) paths in pathways of P.

(b) Visualize a selected subset of pathways in P in full and remainder in collapsed form, for simplicity in visualization.

(c) Using the biochemical networks, **explain** the reasons for blocked (i.e., inactivated) reaction directions in the selected subnetwork.

1.2. A Sample MQL_{AIP} Query Instance and Its Output:

Given:

I. Selected pathways P: Glycolysis, Gluconeogenesis, TCA Cycle, Beta Oxidation, Ketone Body Synthesis, and Fatty Acid Synthesis

II. Selected biological compartment(s): *Mitochondrion*, *Cytosol,* and *Endoplasmic Reticulum* in *Liver*

III. A set C of conditions:

Dietary state(s) and/or physiological condition(s): *Fasting*

Some key metabolite concentration changes (increases/decreases): *lactate↑, alanine↑, triglyceride constituents↑ (i.e., fatty acids↑ and glycerol↑)*

(a) Find activated/inactivated paths in the metabolic subnetwork that includes pathways in P.

(b) Visualize Glycolysis, Gluconeogenesis, and TCA cycle in full and the remainder of the pathways in collapsed form.

(c) Explain the reasons for blocked reaction directions in pathways of P.

Query Result:

(a) Paths with increased flux rate are shown with bold edges in Figure 1.1.

(b) Collapsed (multi-step edges) and full forms of pathways are visualized in Figure 1.1.

(c) Explanations for blocked reaction directions in the selected subnetwork:

- TCA cycle is inhibited due to increased NADH synthesis in Beta Oxidation.
- Fatty Acid Synthesis is inhibited due to: (1) Increased concentration of fatty acids, and (2) Elevated glucagon/insulin ratio in fasting state.
- In Glycolysis, the regulated enzymes (i.e., PFK1,

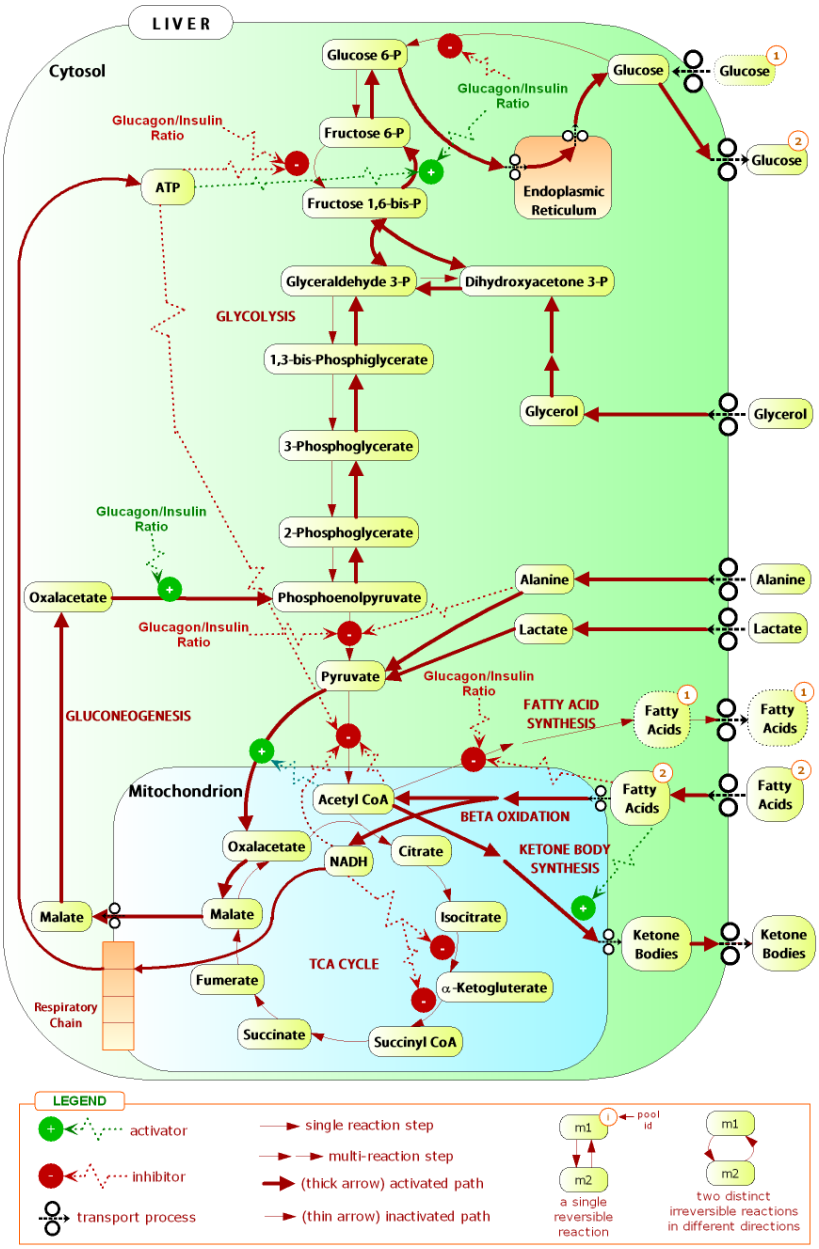

Figure 1.1: A partial human metabolic network in liver

pyruvate kinase, and glucokinase) are inhibited by elevated ATP, alanine, and glucagon/insulin ratio.

- Pyruvate dehydrogenase is inhibited by increased NADH, acetyl CoA, and ATP.

In Section 2, we briefly discuss the related work and our contributions. Section 3 presents an overview of the control mechanisms for the mammalian metabolism. In section 4, we summarize our data and graph models. Section 5 describes the formulation of *MQL* queries. Section 6 elaborates on processing *MQL* queries, and Section 7 concludes.

2. RELATED WORK AND CONTRIBUTIONS

Two studies in the literature focus on the design of query languages for biochemical networks. PQL, the pathway query language, [1] employs a basic graph model where nodes represent metabolites, enzymes, or processes, and edges represent participation of a metabolite in a reaction in different roles, or inhibition/activation relationships between two different enzymes. PQL allows formulation of different query types that include relationship queries (e.g., reactions catalyzed by an enzyme), neighborhood queries, and path queries. bcnQL [2] is another query language with XQuery-like syntax designed for biochemical networks. bcnQL employs an object oriented graph model where nodes and edges have the same semantics as in PQL. bcnQL supports the formulation of almost the same types of queries as PQL with the improvement that bcnQL provides additional capabilities to specify multiple predicates on path queries. Both PQL and bcnQL employ simpler models of metabolic networks, not sufficiently capturing the metabolism: they do not accommodate (i) dynamic behavior of the metabolism under different physiological or dietary conditions, (ii) the compartmentalized structure of the metabolism over tissues and organelles, and (iii) regulatory relationships between pathways. And, neither PQL nor bcnQL provide a capability to specify an initial set of concentration changes on key metabolites to guide query processing, and do not eliminate biologically infeasible query results.

There are also many web-based metabolic databases (e.g., KEGG [3], MetaCyc [5], Reactome [6], PathCase [4], etc.) with query languages. Such data sources serve well for basic database querying via built-in or dynamically constructed queries (e.g., AQI [13], Structured Advanced Query Page and Advanced Query Form [15]). KEGG [3] provides basic keyword search, while MetaCyc [5] and Reactome [6] include advanced query forms. LISP framework in Pathway Tools/BioCyc/MetaCyc provides a custom metabolism-specific query capability [15]. However, the query languages and forms of these data sources also have the same drawbacks listed for PQL and bcnQL. Besides, data models of these data sources do not capture different trigger mechanisms for pathways and/or regulators, distinct contributions of each pathway/ reaction into a particular metabolite pool, and varying occurrences of the same pathway in different tissues.

Qualitative Physics [16] and Qualitative Reasoning (QR) [17, 18] are employed to model systems and environments where measured quantitative information is not available or appropriate to use, but there exists high level (i.e., commonsense) knowledge about the working principles of underlying systems. Studies that adapt qualitative reasoning (or physics) into biochemistry and molecular biology [19, 20] (see [21] for a review) are, at a high level, related to the MQL framework. BioSim [19] is a QR application for simulation of pathways via automated creation of "process" and "object" models based on reactions and their participants (i.e., substrates, products, enzymes), respectively, in a pathway. Then these created modelsof BioSim are executed by a Prolog-based simulation engine which creates a "behavior tree" describing concentration changes for each involved metabolite and enzyme. King et al. [20] extends BioSim's simulation approach, and proposes a model identification framework, where given a qualitative time series set of metabolite measurements, the goal is to identify the structure of the metabolic subnetwork. GenSim [22] is another simulation environment proposed for biochemical systems. In GenSim, users manually construct object (i.e., molecules) and process (i.e., reaction) knowledgebases via a Lisp-based representation scheme, and define preconditions for the processes to be active, as well as their effects, once they are active. There are also many other similar qualitative simulation studies [23-30], which we do not discuss in this paper for the lack of space.

MQL does not directly compare to these simulation works, as it is not designed as a simulation system, but as a query language and its query processing engine. However, MQL employs similar ideas regarding the qualitative reasoning, and (pre)conditions (constraints)

that are checked for reactions to decide their activity status. MQL is different from these simulation studies in that it employs (i) an extensive and detailed metabolism data model, and (ii) fine-tuned biochemical principles, which are not considered by the above-listed works.

2.1. Contributions

In order to process MQL queries, such as the one in Section 1, in a manner that is accurate and consistent with the underlying biochemistry, major biochemical principles that govern the interplay between different metabolic processes should be computationally captured and incorporated into the data model and query processing. Contributions of this paper are as follows.

- Computationally capturing and modeling mammalian metabolic networks by employing the underlying biochemistry principles,
- Design of a metabolism query language (*MQL*), which allows in-depth biochemical queries,
- Development of query processing strategies for *MQL* queries.

Additionally, the work described in this paper expands beyond *MQL*, and constitutes a computational infrastructure for a more informed reasoning of metabolomics data [7].

3. BIOCHEMISTRY-BASED QUERY PROCESSING PRINCIPLES

The processing of an *MQL* query such as the one in Section 1.1 requires an in-depth application of the underlying metabolic principles, which we itemize and review next.

3.1. Substrate Availability

Principle 1. The availability of substrates for a particular pathway is a major driving factor that controls the rate of metabolic processes (i.e., pathways) and biochemical reactions ([9], p. 863).

Example 3.1. Concentration of fatty acids entering liver from blood determines the rate of ketone body synthesis.

Query Processing Rule (QPR) 1: If a substrate concentration increases (decreases), in the absence of other factors, the product concentration also increases (decreases).

Cofactors are small metabolites that bind enzymes and are necessary for biochemical reactions to occur ([9], p. 378). In this work, cofactor in and cofactor out metabolites are considered as specialized substrates and products, respectively.

3.2. Regulation of Key Enzymes

Principle 2. Enzymes are regulated through three major mechanisms: (i) allosteric effects, (ii) covalent modifications, and (iii) expression-level regulation. Regulators either act as inhibitors or activators ([10], p.63). (Regulation through product inhibition is discussed in Section 3.10)

Example 3.2: Citrate activates acetyl-CoA carboxylase which increases fatty acid synthesis.

Query Processing Rule 2: If an activator increases (decreases), the reaction rate increases (decreases). And, if an inhibitor increases (decreases), the reaction rate decreases (increases).

Principle 3. Allosteric effects take place immediately; covalent modification may require minutes; and regulation through gene expression may require hours to days ([10], p. 64).

Query Processing Rule 3: With multiple regulators in effect, consider the one with the quickest mechanism as the "dominant" regulator.

3.3. Regulator Precedence

Principle 4. An enzyme may have multiple regulators which control the enzyme rate with varying degrees of effect. In such cases, usually one of the regulators dominates the other(s) for the final effect (i.e., inhibition or activation) on the enzyme.

Example 3.3: Pyruvate is an activator of pyruvate dehydrogenase (PDH), and acetyl CoA is an inhibitor of PDH. During the fasting state, concentrations of both pyruvate and acetyl CoA increase. However, acetyl-CoA takes the precedence, and the activity of PDH is inhibited.

Query Processing Rule 4: In cases where multiple regulators are in effect, employ the regulator with the strongest effect (highest precedence) on the enzyme, and ignore the other regulators. If no precedence value is available in the database, apply QPR (Query Processing Rule) 3.

3.4. Pathway-Level Regulation

Principle 5. Only a subset of reactions in a pathway is subject to regulation (i.e., regulatory steps or points), and the others simply follow the regulated reactions. In

order for a pathway to have increased flux (or be activated), none of its regulatory reactions should be inhibited.

Example 3.4: In Section 1.1, the regulatory steps of TCA cycle is inhibited by NADH.

Query Processing Rule 5: Do not mark a pathway completely active, if at least one of its regulatory steps (which are pre-captured and made available in the database) is inhibited.

Regulatory steps can further be classified as *rate-limiting* and *commitment* steps.

Example 3.5. Both the rate-limiting and commitment step of Glycolysis is catalyzed by PFK1.

Principle 6. A *rate-limiting* step in a pathway is the slowest step ([9], p. 404).

Principle 7. Once the *commitment step* in a pathway takes place, substrates of a pathway are irreversibly committed to this route in the metabolism ([9], p. 404).

Query Processing Rule 6: A pathway cannot be active if its rate-limiting step is inhibited.

Query Processing Rule 7: If the commitment step of a pathway is blocked (i.e., inactive), then mark the pathway "inactive" as well.

3.5. Metabolite Pools

Principle 8. Due to biological compartmentalization ([9], p. 405) as well as different functional roles [14], a metabolite may have more than one pool.

Example 3.6: Malate is located in both cytosol and mitochondrion. Hence, it has separate pools for each biological compartment that it resides. And, Malonyl CoA is reported [14] to possibly have two pools in cytosol due to distinct functional roles as substrate and regulator.

Query Processing Rule 8: Connect two reactions, r_1 and r_2 in the metabolic network if (i) r_1 and r_2 have at least one shared metabolite m, and (ii) r_1 and r_2 are associated with the same pool of m.

Principle 9. The relative contribution (or consumption) of each metabolic reaction into (from) a specific metabolite pool may differ from one metabolic reaction to another.

Example 3.7: In the fasting state, in liver mitochondrion, the acetyl CoA pool is minimally consumed by TCA Cycle, and the majority of it is used by Ketone Body Synthesis ([9], p. 856).

Query Processing Rule 9: While the direction of change for a particular metabolite pool is computed, take into consideration each contributor's or consumer's rate.

Principle 10. Different pools of the same metabolite may have distinct proportional sizes.

Example 3.8: In the fasting state, the amounts (pool sizes) of different aminoacids released from muscle into blood are not the same, e.g., alanine and glutamine account for 80% of all AAs.

Query Processing Rule 10: Given a metabolite m, if the total size of pools of m with increased (decreased) amounts is larger than the total size of pools of m with decreased (increased) amounts, then overall concentration change of m is considered to increase (decrease).

3.6. Energy State of Cells

Def'n *(Energy Currency Metabolite)*: Certain metabolites serve as carriers of energy in the body. Such metabolites are considered as "energy currency" of cells. Most common energy currency metabolites are NADH, NADPH, $FADH_2$, ATP, and GTP (and their oxidized pairs, i.e., NAD+, NADP+, FAD, AMP, ADP, and GDP, which represent low-energy currencies).

Principle 11. The energy state of the cell can be determined by considering the relative changes in the amounts of high-energy metabolites.

Query Processing Rule 11. Determine the energy state of the metabolism based on the overall change in individual ratios of energy currency metabolite pairs, e.g., $\frac{NADH}{NAD+}$, $\frac{ATP}{AMP}$, and so on.

3.7. Functional Specialization of Biological Compartments

Principle 12. Enzymes in pathways of the human metabolism are highly specialized in terms of biological compartments (e.g., organs, organelles, membranes, etc.) that they reside.

Example 3.10: Urea cycle and ketone body synthesis take place only in liver.

Query Processing Rule 12: Identify enzymes by their biological compartments, and consider their isoforms in different compartments as distinct entities.

Principle 13. Biological compartments are often organized in a hierarchical manner, where one compartment contains another compartment.

Example 3.11: Liver contains mitochondrion which in turn contains mitochondrial matrix.

Query Processing Rule 13: Whenever a compartment is specified in a query, automatically include all of its child (descendant) compartments (if any) in the query.

3.8. Availability vs. Accumulation

Principle 14. Availability of a metabolite m to a pathway or a reaction p means that m is produced, and instantly and fully consumed by p. However, in order for m to accumulate, the rate of m's overall consumption should be less than the rate of its overall production.

Example 3.14: In the fed state, in liver, acetyl CoA is available, through Glycolysis, to Fatty Acid Synthesis and the TCA Cycle, but it does not accumulate. Nevertheless, in the fasting state, acetyl CoA accumulates due to excessive production by Beta Oxidation.

Query Processing Rule 14: Given a metabolite pool m, let ProductionRate(m) be the total production rate of m, ConsumptionRate(m) be the total consumption rate of m. Then, mark m as

- "*severely accumulating*" if ProductionRate(m) > 0 and ConsumptionRate(m) = 0.
- "*accumulating*" if ProductionRate(m) > Consumption Rate(m).
- "*available*" if ProductionRate(m) < Consumption Rate(m) and ProductionRate(m) ≠ 0.
- "*unavailable*" if ProductionRate(m) = 0.

Our metabolite labels above should be interpreted as approximations of the real-world where in reality, due to homeostasis, there is no such thing as "zero" consumption/production rate.

Principle 15. Some reactions require "availability" and some others "accumulation" of a participant metabolite m to assume "substrate availability" (QPR 1) or the regulating effect of m.

Example 3.15: Acetyl CoA is an activator of *pyruvate carboxylase*. In the fed state, acetyl CoA is available, but it does not accumulate. Thus, pyruvate carboxylase is not activated.

Query Processing Rule 15: During query processing, check the "trigger conditions" (i.e., accumulation vs. availability) required by each reaction to make sure that they are satisfied.

3.9. Signatures for Dietary States and Physiological Conditions

Principle 16. In different physiological and dietary states, concentration and/or production rate of certain molecules increase or decrease, which are *signatures* of the corresponding states.

Example 3.16: In the fasting state, glucose levels decrease, and increased amounts of fatty acids and glycerol are released, which leads to increased ketone body production. Also, insulin level decreases and glucagon level increases. Then, signature for the fasting state can be represented as follows: S = {insulin↓, glucagon↑, glucose↓, fatty acids↑, ketone bodies↑, glycerol ↑}.

Query Processing Rule 16: Whenever a user query involves a built-in dietary state or physiological condition predicate, (i) map these predicates to their corresponding signatures, (ii) combine signature changes with the user-provided set of concentration changes, and (iii) over-ride pool contribution rates in the database with those rate changes included in a signature.

3.10. Product Inhibition

Due to similarities in the way they bind to enzymes, products and substrates are in competition to bind to their enzymes, which may lead to inhibition of the corresponding enzyme ([9], p. 405).

Principle 17. As the product concentration of a reaction increases, the reaction rate slows down.

Example 3.17. In the fasting state, due to inhibition of fatty acid synthesis in liver, citrate accumulates, which in turn inhibits the primary reaction (i.e., citrate synthase) that produces it.

In this work, we take a conservative approach on product inhibition by applying it only when a product has no active consumer, i.e., "severely accumulates" (see QPR 14).

Query Processing Rule 17. Whenever a metabolite m is marked as "severely accumulating", mark those reactions that produce m as "inactive".

4. DATA AND GRAPH MODEL

4.1. Data Model

We adopt an object-oriented data model to represent the mammalian metabolism. Objects are structured data types which contain basic types (e.g., string, int, etc.) or other structured data types (i.e., objects) as their fields. We employ the metabolic principles that are summarized in Section 3 as the main motivation, and as

a guide in our modeling effort. Figure 4.1 shows the object definitions and their fields for the essential constituents of the metabolism. Please refer to [8] for an elaborate discussion of the objects and their relation to the principles in Section 3.

4.2. Graph Model

In our graph representation model, compartments are modeled as large "super-nodes" which contain subnetworks of the metabolism, as well as other compartments. In each subnetwork, nodes represent metabolite pools. Reactions are represented as hyper-edges, which connect multiple end-points (i.e., substrates and products). Regulation is represented by an edge between a metabolite pool (i.e., a regulator) and a hyper-edge (i.e., a reaction).

Pathway { name: string
reactions: [ReactionStep]
(rateLimitingSteps: [ReactionStep])
(committedStep: ReactionStep)
substrates: [MetabolitePoolLink]
(products: [MetabolitePooLink])
(cofactorsIn:[MetabolitePoolLink])
(cofactorsOut:[MetabolitePoolLink])
}
ReactionStep {reaction: Reaction
compartment: Compartment
(direction: <forward | backward>)
}
Reaction { name: string
substrates: [MetabolitePoolLink]
products: [MetabolitePoolLink]
(cofactorsIn:[MetabolitePoolLink])
(cofactorsOut:[MetabolitePoolLink])
(enzymes: [EnzymeInstance])
(inhibitors: [Regulator])
(activators: [Regulator])
isTransportProcess: boolean
isReversible: Boolean
}
MetabolitePool{ metabolite: Metabolite
compartment: Compartment
(name: string)
(size: int)
(parent: MetabolitePool)
}
MetabolitePoolLink {
pool: MetabolitePool
rate: float
triggerCondition: listed in QPR 14
stoichiometry: float
}
Metabolite { name: string
type: string
defaultPools: [MetabolitePool]
}

Compartment { name: string
(size: int)
(parent: Compartment)
(transportProcesses: [Reaction])
type:<tissue | organelle |
membrane>
}
Enzyme{ name: string }
EnzymeInstance{ enzymeId: int
compartment: Compartment
}
DietaryState { name: string
concentrationChanges:
[ConcentrationChange]
}
ConcentrationChange {
pool: MetabolitePool
changeDirection: string
}
PhysiologicalCondition extends
DietaryState
{
(rateChanges: [RateChange])
}
RateChange {
poolLink: MetabolitePoolLink
rateChangeAmount: float
}
EnergyCurrencyMetabolite
extends **Metabolite**{
peer: EnergyCurrencyMetabolite
chargeStatus: <high | low>
}
Regulator {(regulator:
MetabolitePool)
triggerCondition: listed in QPR 14
type: <allosteric | covalent |
expressionRegulation>
(precedence: int)
(regulatorRatio: $\frac{MetabolitePool\ 1}{MetabolitePool\ 2}$)
}

Notation: [ ] denotes an array of objects, < a | b | c > denotes an enumeration, and ( ) represents optional fields

Fig. 4.1: Object Data Model

5. QUERY SPECIFICATION

In this section, we discuss the formulation of MQL_{AIP} queries. Please see [8] for the specification of the remaining types of queries that can be formulated in *MQL*.

MQL_{AIP} queries involve finding activated/inactivated paths in a particular subnetwork of the metabolism in specified biological compartments under particular dietary or physiological states with a given set of concentration changes of key metabolites (see Section 1.1).

We adopt an SQL-like [11] database query specification scheme for MQL_{AIP} queries where the *select* clause defines the output, the *from* clause defines the objects/relations, and the *where* clause describes additional predicates such as compartment, dietary state, and so on. The generic query template is formulated as shown in Figure 5.1 where the notation is as follows.

- Names in italics refer to database values, e.g., *cytosol* or *TCA-cycle.*
- Names in regular fonts (i.e., non-italic and non-bold-face) refer to variables, e.g., P1, C1, etc.
- Bold-face words refer to keywords of the query language, e.g., **select**, **paths**, etc.
- [a | b | c] denotes *exactly one of* (a *or* b *or* c).
- { a | b | c} denotes *exactly one or zero of* (a *or* b *or* c).
- The parenthesis, ( ), has no particular meaning, and is used solely for grouping purposes.
- ".." denotes zero or more repetitions.
- "*" stands for the quantifier "all" as in the standard SQL specification [11].
- If, for a particular field, nothing is specified as part of the query, then default selections are assumed.

Default selections are marked as "(default)" in the query template.

- . notation in compartments is used to specify root-to-node path expressions (as in the path expressions of object-oriented query languages [12]) in the compartment hierarchy.
- Visualizing pathways in full/collapsed forms or providing additional explanations about activated and/or inactivated reactions are specified as optional separate clauses.
- *a-set-of-conditions* refers to a set of database physiological conditions, such as *diabetes.*
- *a-set-of-metabolite-concentration-changes* refers to a set of database metabolite concentration changes, each in a specific compartment.
- *a-subset-of-input-pathways* refers to a subset of the pathway variables specified in the from clause.
- The concentration change direction symbols ↑ and ↓ are replaced with ∧ and ∨, respectively.

select {**activated** | **inactivated** | * (default)} **paths**
from pathways *Pi1* P1 **in** *Ck1*{.*Ck2*..*Ckm*} C1{, *Pi2* P2
in *Cr1*{.*Cr2*..*Crm*} C2, .., *Pin* Pn
in *Ct1*{.*Ct2*..*Ctm*} Cm }
where {**dietaryState** = [**fasting** | **after a meal devoid of AAs** | **after a meal devoid of carbs** | **exercise** | **well-fed**]}
{**and physiologicalCondition** = *a-set-of-conditions*}
{**and concentrationChanges** = *a-set-of-metabolite-concentration*-changes}
{**visualize** {*a-subset-of-input-pathways in P1 [, P2, .., Pn]* | * (default)} {**as collapsed** (default) | **as full**}}
{**explain**{ **inactivated** (default) | **activated** | **regulated**} **reactions** [**in** *a-subset-of-pathways* | *]}

Fig. 5.1: MQL_{AIP} query template

Example 5.1: The following query represents the MQL_{AIP} specification of the English query which is discussed in Section 1.1.

select * paths
from pathways Glycolysis P1 **in** liver.cytosol C1,
Gluconeogenesis P2 **in** liver.cytosol C1,
TCA-Cycle P3 **in** liver.cytosol.mitochondrion C2,
Beta-Oxidation P4 **in** liver.cytosol.mitochondrion C2, Keone-Body-Synthesis P5 **in** liver.cytosol.mitochondrion C2, Fatty-Acid-Synthesis P6 **in** liver.cytosol C1
where dietaryState = fasting
and concentrationChanges = {*lactate* ∧ **in** *blood*, *alanine* ∧ **in** *blood*, *fatty acids* ∧ **in** *blood*, *glycerol* ∧ **in** *blood*}
visualize P1, P2, P3 **as full**
explain inactivated reactions in *

6. PROCESSING OF MQL_{AIP} QUERIES

In this section, we discuss the query processing of MQL_{AIP} queries. We employ the biochemical query processing rules (QPRs) of Section 3. Related QPRs are often referenced in parentheses. The query processing employs an auxiliary data structure, called *dependency graph*. We first define dependency graphs, and then move onto the discussion of the query processing stages.

Def'n *(Dependency Graph)*: A dependency graph G(V, E) consists of a set V of vertices and a set E of edges, where nodes in V correspond to distinct pathways, and a directed edge e(p_1, p_2) represents that pathway p_1 (i.e., e.*source*) is dependent on pathway p_2 (i.e., e.*destination*).

6.1. Stages of the Query Processing

Input to this type of queries are a subnetwork of the metabolic network as defined by a set of pathways in specific compartments, a dietary state condition, a physiological condition, and a set of initial concentration changes. The query processing has three stages.

(a) Stage 1 ***(Query Compilation)***

1.1. Convert dietary state and physiological condition predicates in the query into their signature concentration change sets (*QPR 16*). Let S be the union of such concentration change sets.

1.2. Let C be the user-specified concentration changes in the where clause of the query. Then, let U = S ∪ C. Mark those pools in U with increased concentrations as "accumulating", and those with decreased concentrations as "unavailable". Mark all other metabolite pools "unavailable".

1.3. Expand user-provided set P of pathways with additional pathways that connect user-provided metabolites to the pathways in P (e.g., Pyruvate to Acetyl CoA pathway in Fig. 1.1).

1.4. Let M be the compartment set (and their descendants) specified in the query. (*QPR 13*)

1.5. Build the *query subnetwork* with the pathways

in P and compartments in M (*QPR 8, 12, 13*).

1.6. Create three empty pathway sets, P^{active}, $P^{inactive1}$, $P^{inactive2}$, and P^{sink}. Initialize $P^{inactive1} = P$.

In stages 2 and 3, we record the regulation information of pathways. Since it does not involve much complexity, for brevity, we will not refer to keeping track of the regulation information.

(b) Stage 2 ***(Identifying completely active pathways)***

In this stage the set P^{active} of completely active pathways are identified in two substages.

b.1. **Expansion**: Expansion is an iterative process, where, in each iteration, the set of active and/or inactive pathways are expanded based on the availability of substrates and regulators.

2.1. Identify subset P_S of pathways in $P^{inactive1}$ such that, for each pathway p in P_S, at least one of its substrates is available with a matching trigger condition (*QPR 1, 14, 15*).

2.2.$P^{active} = P^{active} \cup P_S$ ($\cup$ denotes set union, $\cap$ denotes set intersection, and "–" denotes set difference.

$$P^{inactive1} = P - (P^{active} \cup P^{inactive2} \cup P^{sink})$$

2.3. Update status of pools that are involved in pathways in P^{active} (*QPR 14*).

2.4. If content of P^{active} has changed in step 2.2, go to step 2.1 for another iteration.

b.2. **Shrinking:** Shrinking is also an iterative process where the set of active pathways is shrunk based on the accumulation or availability of energy currencies, substrates, and cofactors.

2.5. Consider pools for energy currency metabolites, cofactors, and other regulators (*QPR 1, 2, 3, 4, 11, 14*). Then, consider the reactions that are regulated by these pools (*QPR 2-7*). Next, according to the nature of regulation (*QPR 15*), mark pathways involving such reactions "inactive" (*QPR 5-7*), and move them into $P^{inactive2}$. Update the marks of metabolite pools. Next, reconsider substrate availability for pathways in P^{active}, and move those with non-matching trigger conditions from P^{active} to $P^{inactive2}$ (*QPR 1, 15*). Finally, apply product inhibition (*QPR 17*), if applicable. Update metabolite pool marks.

2.6. If content of $P^{inactive2}$ has changed, go to step 2.5. Otherwise, continue with the next step.

2.7. Initialize a dependency graph G(V=P, E=∅). Identify each pathway $p_1 \in P^{inactive2}$ such that p_1 was put into $P^{inactive2}$ due to a non-matching trigger condition on a metabolite pool m, but p_1 is not inhibited anymore due to change on the marking of m. Add an edge $e(p_1, p_2)$ in G for each $p_2 \in P^{inactive2}$ where p_2 is a consumer/producer of m. Next, identify cycles in G, move members of each cycle into P^{sink}. For each edge e that is not part of a cycle in G, move e.*source* into $P^{inactive1}$.

2.8. If content of P^{active} or $P^{inactive1}$ has changed in step 2.5, go to step 2.1, else continue with step 3.1.

(c) Stage 3 ***(Identifying partially active pathways)***

The last query processing stage focuses on identifying partially active pathways. It builds upon the general idea that there will be an active flux through non-regulated reactions as long as their substrates are available (or accumulate, depending on the trigger condition) given that there is no "product inhibition" (*QPR 17*).

3.1. Mark all reactions in active pathways as *active*, with one exception: mark all inhibited reactions (by a direct inhibitor) as *inactive*. Let the active reaction set be R^{active}. Finally, mark all other reactions as *maybe-active* (i.e., *unknown*). Update metabolite pool marks accordingly (*QPR 14*). During the application of *QPR 14*, consider *maybe-active* reactions as *inactive*.

3.2. Identify pathway pairs (p_1, p_2) of pathways where (i) $p_1 \in P^{active}$, $p_2 \in P^{inactive}$, (ii) p_1 and p_2 have a shared metabolite m (as product and substrate) (*QPR 8*). Form triplets (p_1, p_2, m).

3.3. For each triplet (p_1, p_2, m) from step 3.2, starting with reaction(s) that consume m in p_2, navigate the query subnetwork reaction-by-reaction by following substrate/product relationship, and employ the following actions (let r be the currently traced reaction):

(a) If r is an *active* reaction, stop the traversal on this path in the query subnetwork.

(b) If r is an *inactive* reaction, stop the traversal (*QPR 17*), trace back while marking each reaction *inactive* on the way back until (i) the starting metabolite m, or (ii) the first metabolite with *maybe-active* consuming reactions (whichever is encountered first).

(c) Otherwise, mark r *active*, if at least one of its substrates is available with a matching trigger condition (*QPR 1, 14*). Continue employing the same considerations ((a) through (c)) iteratively with the next set of reactions that follow r in the query subnetwork.

3.4. Similar to step 2.6, consider pools of

metabolites that participate in *active* reactions as substrates, cofactors, or regulators. Accordingly, if there are inhibited reactions or ones with unavailable substrates (according to its trigger condition), mark them as *inactive*. Apply product inhibition (*QPR 17*) if applicable. Iterate over this step, if new *inactive* reactions are identified.

3.5. Mark all the remaining *maybe-active* reactions *inactive*, and add the *active* ones into R^{active}.

6.2. A Complete Example (Processing the MQL Query in Example 5.1)

Stage 1:

- Converting the fasting stage into its signature concentration change set: S = {insulin↓, glucagon↑, glucose↓, fatty acids↑, ketone bodies↑, glycerol ↑}. Figure 1.1: Each hormone (e.g., insulin, glucagon) has a single pool (not shown in Fig. 1.1); glucose in S refers to its pool #1 in blood; fatty acids refers to its pool #2 in blood; and, others have single pools in blood in Fig. 1.1.
- User-provided concentration changes are mapped to their default pools in blood. Figure 1.1 has only a single pool for each of lactate, alanine, and glycerol in blood (i.e., default pools). For fatty acids, the default pool in blood is pool #2.
- Take the union of user-provided set of metabolite pools and those in S, U = {insulin↓, glucagon↑, glucose↓, fatty acids↑, ketone bodies↑, lactate↑, alanine↑}, and update their pool marks accordingly.
- Expand user provided set P of pathways with "connection pathways". Hence, P = {*Glycolysis*, *Gluconeogenesis*, *TCA-Cycle, Beta-Oxidation, Ketone-Body-Synthesis, Fatty-Acid-Synthesis, Respiratory Chain, Glycerol2Dihyroxyacetone3-P, Alanine2Pyruvate, Lactate2Pyruvate, Pyruvate2AcetylCoA*}. And, $P^{inactive}$ = P.

Stage 2:

- *Iteration 1:*
  1. P_S = {*Beta-Oxidation, Glycerol2Dihyroxyacetone3-P, Alanine2Pyruvate, Lactate2Pyruvate*}
  2. P^{active} = {*Beta-Oxidation, Glycerol2Dihyroxyacetone3-P, Alanine2Pyruvate, Lactate2Pyruvate*}
     $P^{inactive1}$ = {*Glycolysis*, *Gluconeogenesis*, *TCA-Cycle, Ketone-Body-Synthesis, Fatty-Acid-Synthesis, Pyruvate2AcetylCoA, Respiratory Chain*}
  3. Update pools: Acetyl CoA↑, NADH↑, Pyruvate↑, Dihydroxyacetone 3-P↑.
  4. Since P^{active} has changed, perform another iteration over steps 1 through 4.
- *Iteration 2:*
  1. P_S = {*Gluconeogenesis, TCA-Cycle, Fatty-Acid-Synthesis, Pyruvate2AcetylCoA, Ketone-Body-Synthesis, Fatty-Acid-Synthesis, Pyruvate2AcetylCoA, Respiratory Chain*}
  2. P^{active} = {*Beta-Oxidation, Glycerol2Dihyroxyacetone3-P, Alanine2Pyruvate, Lactate2Pyruvate*, ***Gluconeogenesis***, ***TCA-Cycle, Ketone-Body-Synthesis, Fatty-Acid-Synthesis, Pyruvate2AcetylCoA, Respiratory Chain***}
     //**bold** ones are newly added in this iteration
     $P^{inactive1}$ = {*Glycolysis*}
  3. Update pools: Glucose↑, Ketone Bodies↑, NADH↑.
  4. Since P^{active} has changed, perform another iteration over steps 1 through 4.
- *Iteration 3:*
  1. P_S = { *same as the previous iteration*}
  2. P^{active} = // the same as in iteration 2.
     $P^{inactive1}$ = {*Glycolysis*}
  3. Update pools: no changes.
  4. Since P^{active} has not changed, go to step 5.
  5. Production rate of the energy currency metabolite NADH by Beta-Oxidation and TCA-Cycle is much more than its consumption rate through Respiratory Chain. Hence, NADH pool accumulates. NADH inhibits two rate limiting steps of TCA-Cycle. Fatty Acid Synthesis and Pyruvate2AcetylCoA are also inhibited by fatty acids and glucagon. Thus,
     P^{active} = {*Beta-Oxidation, Glycerol2Dihyroxyacetone3-P, Alanine2Pyruvate, Lactate2Pyruvate*, *Gluconeogenesis*, *Ketone-Body-Synthesis, Respiratory Chain*}

$P^{inactive2}$ = {***TCA-Cycle, Fatty-Acid-Synthesis, Pyruvate2AcetylCoA***}

6. Since $P^{inactive2}$ has changed, another iteration over step 5 is required, but nothing changes.
7. Since P^{active} has changed in step 5, iteration 4 is performed, but it does not change the set of active/inactive pathways. Hence, we do not present iteration 4, here.

Stage 3: There is no partially active pathway in this example.

Query Result: Please see Section 1.1 for query results. Step 6 also lists the active pathways.

6.3. Handling Inconsistent Input to MQL_{AIP} Queries

It is possible that the user-provided concentration change statements in a query may be inconsistent with respect to the activated/inactivated set of pathways included in the query result. In characterizing such types of query input inconsistencies, we employ the *closed world assumption* [11].

Def'n *(Closed World Assumption)*: Given (i) a metabolite pool m, and (ii) a query subnetwork N, let (a) R be the set reactions in N, (b) C(m) be the consumers of m, and (c) P(m) be the producers of m. Then, C(m) $\subseteq$ R and P(m) $\subseteq$ R always hold.

Now, we are in a position to formally define query input inconsistency based on the *Closed World Assumption* and the Query Processing Rule 14.

Def'n *(Inconsistent Query Input)*: Given an MQL_{AIP} query *Q*, let *C* be the set of metabolite concentration change pairs (m_i, c_i) where m_i is a metabolite pool, and c_i is a concentration change statement (i.e., "increase" or "decrease"), that are either provided directly by the user, or obtained from the signature of a dietary state or physiological condition included in the query. Then, *Q* is called an *inconsistent query* if there is at least one concentration change pair (m_i, c_i) $\in$ *C* such that, in the query subnetwork, after the query is processed, (i) m_i is marked as *unavailable*, and c_i = increase, and/or (ii) m_i is marked as (*severely*) *accumulating* in the query result, and c_i = decrease.

At the end of Stage 3 of the query processing, the above definition is employed to determine if the query is inconsistent. Inconsistent queries return empty result sets. However, as an explanation, users are also provided with those particular metabolite-concentration change pairs that render the query inconsistent.

6.4. Discussion

In this section, we present a brief discussion on the termination behavior of our query processing algorithm which contains looping structures among/within different steps. It is crucial that the algorithm does not get into infinite loops. In the algorithm, since there is no loop that spans over multiple query processing stages, each stage can be analyzed independently.

Stage 1 does not contain any looping structure.

In Stage 2, pathways are moved between three different sets that may potentially lead to infinite loops: $\mathbf{P^{inactive1}}$ → Steps 2.1 .. 2.4 → P^{active} → Steps 2.5 .. 2.6 → $P^{inactive2}$ → Step 2.7 → $\mathbf{P^{inactive1}}$. If a pathway p is continuously circulated through these three sets, then the algorithm never terminates. Such cases may happen when there is a set of pathways which are inter-dependent on each other in a cyclic manner through regulatory relationships. Hence, such pathways with cyclic interdependency should be eliminated from consideration in Stage 2. Construction of a dependency graph in step 2.7 and removal of pathways (into P^{sink}) that form cycles is integrated into the algorithm to prevent such infinite looping cases. Hence, we have Lemma 6.1 (proofs are omitted for brevity).

Lemma 6.1: *Stage 2 of the algorithm always terminates, i.e., infinite loops never occur.*

Finally, in Stage 3, the only looping structure takes place in step 3.4 which iterates over itself. This step performs backtracking due to a product inhibition at intermediate steps of a pathway. Hence, it only expands the set of inactive reactions, and does not manipulate the set of active reactions. Therefore, the number of iterations is bounded by the total number of reactions in the query subnetwork. Hence, we have Lemma 6.2.

Lemma 6.2: *Stage 3 of the algorithm always terminates, i.e., infinite loops never occur.*

7. CONCLUSION

In this paper, we have presented the design and query processing of a metabolism query language, MQL, which allows users to specify detailed metabolism queries. Our approach is faithful to the underlying biochemistry, and completely guided by the metabolic principles. MQL is presently being implemented and integrated into the Metabolomics Analysis Workbench [7] of PathCase [4].

References

1. Leser, U. 2005. A query language for biological networks. ECCB/JBI 2005: 39, *Bioinformatics*: 21 (Suppl 2): ii33-ii39.
2. Yang, H, Sunderraman, R, Tian, H. 2008. bcnQL: A Query Language for Biochemical Network. *IEEE International Conference on Bioinformatics and Biomedicine, BIBM '08*.
3. Kanehisa M et al. 2006. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res* 34:D354–7.
4. Elliott, B., Kirac, M., Cakmak, A. et al. 2008. PathCase: Pathways Database System. *Bioinformatics* 24(21): 2526-2533.
5. Caspi, R et al. 2006. MetaCyc: a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Res* 34 (Database issue):D511–16.
6. Joshi-Tope G et al. 2005. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Res* 33: D428–32.
7. Cakmak, A., Dsouza, A., Hanson, RW, Ozsoyoglu, ZM, Ozsoyoglu, G. 2009a. Analyzing Metabolomics Data for Automated Prediction of Underlying Biological Mechanisms. *BMC Bioinformatics* (*Submitted*).
8. Cakmak, A, Hanson, RW, Ozsoyoglu, G. 2009b. Querying Mammalian Metabolism under Different Physiological Conditions. *Journal of Bioinformatics and Comp. Biology (Invited for April 2010 Issue)*.
9. Devlin, TM. 2006. Textbook of Biochemistry with Clinical Correlations, Sixth Edition. Hoboken, NJ, *John Wiley & Sons*.
10. Champe, PC, Harvey, RA, Ferrier, DR. (2007). Lippincott's Illustrated Reviews: Biochemistry. *Lippincott Williams & Wilkins*; Fourth Edition.
11. Ramakrishnan, R., Gehrke, J. Database Management Systems. Third Edition. *McGraw-Hill*, 2003.
12. Bertino, E. Negri, M. Pelagatti, G. Sbattella, L. Object-oriented query languages: the notion and the issues. *IEEE Transactions on Knowledge and Data Engineering*. 4(3): 223-237, June 1992.
13. Elliott, B., Mayes, S., Cakmak, A. et al. 2009. Advanced Querying Interface for Biochemical Network Databases. *BMC Bioinformatics* (*Submitted*).
14. Saggerson, David. Malonyl-CoA, a Key Signaling. Molecule in Mammalian Cells. *Annu. Rev. Nutr*. 2008. 28:253-72.
15. Krummenacker, M., Paley, S., Mueller, L., Yan, T., Karp, PD. Querying and Computing with BioCyc Databases, *Bioinformatics* 21:3454-5, 2005.
16. Weld, DS, De Kleer, J. (1990). Readings in Qualitative Reasoning About Physical Systems. *Morgan Kaufmann*, San Mateo, CA.
17. Forbus, KD. Qualitative process theory. *Artificial Intelligence*, 24:85–168, 1984.
18. Kuipers, B. Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge. *MIT Press*, 1994, Cambridge, MA.
19. Heidtke, KR, Schulze-Kremer, S. BioSim - A New Qualitative Simulation Environment for Molecular Biology. *In Proc. of 6th Int. Conf. Intell. Syst. Mol. Biol. (ISMB98)*, pp. 85–94.
20. King, RD, Garrett, SM, and Coghill, GM. On the use of qualitative reasoning to simulate and identify metabolic pathways. *Bioinformatics* 2005 21: 2017-2026.
21. De Jong, H. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology* 2002;9(1):67-103.
22. Karp, PD. (1993). A qualitative biochemistry and its application to the tryptophan operon. In *Artificial Intelligence and Molecular Biology*, pp. 289 – 324.
23. Meyers, S. and Friedland, P. (1984). Knowledge based simulation of genetic regulation in bacteriophage lambda. *Nucleic Acids Research* 12(1):1–9.
24. Thomas, R. (1991). Regulatory Networks Seen as Asynchronous Automata: A Logical Description. *Journal of Theoretical Biology*, 153:1 – 23.
25. Brutlag, D. L., Galper, A. R., and Millis, D. H. (1991). Knowledge-Based Simulation of DNA Metabolism: Prediction of Enzyme Action. *Computer Applications in Biosciences*, 7(1):9 – 19.
26. Shimada, T., Hagiya, M., Arita, M., Nishizaki, S., and Tan, C. (1995). Knowledge-based Simulation of Regulatory Action in Lambda Phage. *Journal of Artificial Intelligence Tools*, 4(4):511–524.
27. Clancy, DJ, Kuipers, B. (1997). Model decomposition and simulation: A component based qualitative simulation algorithm. *14th National Conference on Artificial Intelligence (AAAI-97)*.
28. Karp, P. D. and Mavrovouniotis, M. M. (1994). Representing, analayzing, and synthesizing biochemical pathways. *IEEE Expert*, 9(2):11 – 22.
29. Kazic, T. (1993). Reasoning about biochemical compounds and processes. *In Proceedings of the International Conference on Bioinformatics, Supercomputing and the Human Genome Project*, pages 35 – 49, Singapore. World Scientific.
30. Kazic, T. (1993). Representation, reasoning and the intermediary metabolism of E. coli. In *Proceedings of the Hawaii International Conf. on System Sciences*, volume 1, pages 853 – 862.