Computational Interpretation of Metabolomics Measurements: Steady-State Metabolic Network Dynamics Analysis

Ali Cakmak ali.cakmak@case.edu
Xinjian Qi xinjian.qi@case.edu
A. Ercument Cicek aec51@case.edu
Gultekin Özsoyoğlu tekin@case.edu

Department of Electrical Eng. and Computer Science
Case Western Reserve University

ABSTRACT

With recent advances in experimental technologies, the number of metabolites measured in bio-fluids of organisms has markedly increased. Given a set of measurements, a common metabolomics task is to identify the metabolic mechanisms that lead to changes in the concentrations of given metabolites, and interpret the metabolic consequences of the observed changes in terms of physiological problems, nutritional deficiencies, or diseases.

This paper presents the SMDA (steady-state metabolic network dynamics analysis) technique and its computational performance limits using a mammalian metabolic network database. The query output space of the SMDA tool is exponentially large in the number of reactions of the network. However, (i) larger numbers of observations exponentially reduce the output size, and (ii) exploratory search and browsing of the query output space allows users to mine and search for what they are looking for.

Categories and Subject Descriptors

J.3 [Computer Applications]: Life and Medical Sciences – *biology and genetics.*

General Terms

Algorithms, bioinformatics.

Keywords

SMDA, Metabolomics, Metabolic Network, *In Silico* Analysis.

1 INTRODUCTION

Currently, metabolomics data analysis necessitates a time-consuming, extensive, and manual cross-referencing of metabolic pathways, in order to critically evaluate the measurements data. Recently, an excellent novel *In Silico* approach (IOMA) that integrates metabolomics data with a metabolic network model, and infers metabolic fluxes is proposed [1]. However, IOMA (a) requires many information (e.g., availability of the stoichiometry matrix of the network, dissociation constants, enzyme turnover rates, mass balance constraints, flux capacity constraints, etc.), and (b) infers a *single* network state with all the computed metabolic fluxes.

In this paper, we propose a much simpler database-enabled and graph-traversal-based technique, called SMDA (***S**teady-state **M**etabolic network **D**ynamics **A**nalysis*), that infers all allowable states of the network. Given a set of bio-fluid (e.g., blood) and tissue-based metabolite concentration measurements at steady-state, SMDA answers the question of "what type of alternative steady-state metabolic network activation/inactivation scenarios exist, given the observed measurements?" In more detail, SMDA takes as input user's (i) metabolomics data, (ii) metabolic sub-network, selected from a metabolic network database already available to users, and produces a set of possible alternatives for active/inactive metabolic sub-networks.

SMDA can be viewed as both a constraint- and rule-based approach. It is constraint-based [2, 3, 4] in that it uses conditions (pre-stored in its database) to locate all "allowable states" [5] of the reconstructed metabolic network model (pre-stored in its database). And, it is rule-based in that its graph-expansion and merge strategies employ a number of biochemistry rules to capture the underlying metabolic biochemistry as much as possible. Advantages of SMDA include its ease of use and simplicity; it is designed as a "first-step" and 'online" tool for wet lab researchers (a) to evaluate their hypotheses about observed measurements, and (b) to be used for "what if" types of questions (i.e., knowledge discovery). The disadvantages of SMDA include: (a) it returns only two flux values for a reaction, namely, 0 (inactive), and 1 (inactive); (b) as is the case with other techniques that return "all allowable states" [2], it is inherently exponential. However, the computational performance of SMDA is acceptable for networks with up to 60 reactions (with some paths/pathways abstracted into "abstract reactions"; see supplement [8] and Section 4). SMDA is implemented, and available on the web as an online tool [6], as part of PathCase family of applications [7].

1.1 SMDA Overview

Prior Preparation. SMDA database has a fully hierarchical and compartmentalized metabolic network, i.e., one with tissues, organelles, etc. And, the steady-state "activation conditions" (or, the *ACT condition set*) for each reaction and transport process to be active (i.e., flux: 1) are characterized a priori, saved in a database, and used during query-time analysis. Initially, the status values of all reactions and all metabolite pools in the metabolic network are *Unknown*.

Query-time Analysis. At query time, the user chooses a metabolic sub-network to query. SMDA takes the observed metabolite set and the selected smaller sub-network as input, and executes the following steps.

- *Initialization.* (i) For each bio-fluid-based metabolite observation, it identifies whether its transport processes are active or not (by checking, for each transport process, whether all conditions in its ACT set are satisfied or not). (ii) For each tissue-based metabolite observation, it derives its metabolite pool label, which is one of *Unavailable, Available, Accumulated,* or *Severely Accumulated.*
- *Expansion: Metabolic Network Traversal and Active-Inactive Reaction Assessment.* Starting with active/inactive transport processes and tissue-based observed metabolites,

ACM-BCB '11, August 1-3, Chicago, IL, USA

and continuing with metabolic reactions in tissues, SMDA locates iteratively those reactions with satisfied or unsatisfied ACT condition sets, and marks (i) those reactions whose ACT conditions are completely satisfied as *Active*, and (ii) those reactions whose ACT conditions contain at least one unsatisfied ACT condition as *Inactive* (i.e., flux is 0).

The above summarized query-time analysis creates and iteratively expands multiple possible metabolic sub-graphs, called *Active-Inactive Graphs* (G_{AI}), where, in each G_{AI} graph, the status of each reaction, and the label of each metabolite pool is clearly marked (i.e., no reactions or metabolite pools with "*Unknown*" status/label). The result is a set of G_{AI} graph sets where each G_{AI} graph set specifies one distinct alternative steady-state activation/inactivation scenario for the metabolic network. An alternative output to G_{AI} graphs is *R-graphs* where an R-graph is a G_{AI} graph without metabolite pool labels. We give an example.

Example 1.1. Assume that the user selects *Catabolism of Cysteine* in liver as the metabolic sub-network to be queried, and has three observed metabolite measurements in cytosol: O_2 as 80mM/L (we assume that O_2 is "estimated" as it is very difficult to measure O_2 in tissue of intact organ), *cysteine* as 60 μM/L, and SO_3 (3-sulifino-L-Alanine) as 80 μM/L. Assume that the database conditions state that, in Liver cytosol, "O_2 is marked as *Available* if it is in between [1, 100]mM/L", "*cysteine* is marked as *Available* if it is in between [1, 100] μM/L", and "SO_3 is marked as *Available* if it is in between [1, 100] μM/L". Thus, the SMDA initialization step concludes that O_2, *cysteine*, and SO_3 are all *Available.* And, the execution of the expansion step as summarized above concludes that there is only one R-graph with only one G_{AI} graph in the output of the query, as shown by the (actual) SMDA output of Figure 1.1.

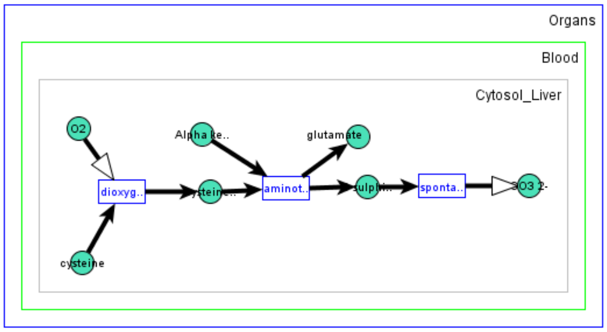

Figure 1.1. SMDA result as a single G_{AI} graph

In summary, given metabolomics observations and a query sub-network, *SMDA* locates all possible alternative active-inactive scenarios on the sub-network. This approach provides compact and complete steady-state views of possible metabolism dynamics as independent and alternative snapshots in the form of user-friendly visual steady-state views of the metabolic network. There are two issues: (i) prioritizing and ranking different alternatives--not discussed in this paper; please see the supplement [7] for a number of ranking mechanisms. (ii) What happens when, for a large sub-network, there are many alternative G_{AI} graphs? As a response to this issue, SMDA allows for an exploratory search of the resulting G_{AI} graphs. That is, an "interactive query" execution takes place where, as a response to the query, the user is given the total number of "possible results" (i.e., G_{AI} graphs), and, is then prompted to choose and view different G_{AI} graphs in the output with respect to participating metabolites and enzymes. For example, the user is told, say, that *Pyruvate dehydrogenase* is active in two R-graphs and inactive in four R-graphs, and is given the option of viewing only the first two, or the latter four, or all six R-graphs. We refer to this process as an "*exploratory search and browsing*" of the SMDA query output search space.

The observation set of example 1.1 is available as "Sample Observation 0" on the web [6]; and, running the SMDA Tool with Sample Observation 0 produces the results of example 1.1. SMDA tool, an evolution of OMA Tool [9], is currently being beta-tested in cystic fibrosis metabolomics data analysis.

This paper is organized as follows. Section 2 specifies a complete condition-based model of the metabolic network behavior. We (i) list the assumptions of our model and define the notion of (quasi-) steady-state for the metabolic network, (ii) introduce the notion of metabolite pool label identifiers, (iii) employ a three-valued logic to specify metabolite pool label conditions and *Activation Condition Sets* for reactions as well as transport processes, (iv) list transport process rules, and, finally, (v) specify a number of basic biochemistry-based rules. Due to space constraints, in the supplement [8], we present the SMDA algorithm with G_{AI} (R-) graph initialization, expansion, and merge steps. The algorithm iteratively constructs the *G_{AI} Generation Hierarchy* where, when it terminates, each leaf node of the hierarchy contains one possible activation/inactivation scenario within the query sub-network. Also, in the supplement, we specify three different alternative expansion strategies for the expansion step. Section 4 presents a brief computational performance evaluation of the SMDA tool by using PathCase-MAW mammalian metabolic network database. Section 5 lists future work.

2 CONDITION-BASED MODELING

2.1 Assumptions and Terminology

We make the following assumptions about our environment.

- Complete metabolic network is pre-captured and available in a metabolic network database.
- The metabolic network database models tissue-level compartmentalization; that is, it is a multi-tissue and a multi-compartment (e.g., *cytosol, mitochondrion,* etc.) environment.
- The metabolic network is "sound" in the sense that all metabolites that are not in bio-fluids are both produced by (i.e., are a product of) at least one reaction *and* consumed by (i.e., are a substrate of) at least one reaction.
- Initially, we label each unmeasured metabolite pool size with the identifier "*Unknown*". During query-time analysis, the labels may change into one of "*Unavailable*", "*Available*", "*Accumulated*", or "*Severely accumulated*". The reason for nonquantitative labeling (as opposed to numerical size values) is that this paper does not employ quantitative pool size estimation techniques, discussed in more detail in Section 2.2.
- No a priori knowledge of the size of each metabolite pool is assumed, except for measured metabolites.
- Given a reaction r, and a metabolite m as a substrate, co-factor-in, activator (product, co-factor-out, inhibitor) of r, the knowledge of the lowest (highest) metabolite pool size label of m at steady-state for m to activate (inhibit) a reaction so that r is "active" ("inactive"), is assumed to be available. This is discussed more in Section 2.4 below.
- The organism (represented by its metabolic network database) is queried when it is at a *steady-state* for a time interval *T*. Steady-state is defined in terms of two properties:

a. *Production-Consumption Rate Equality (PCRE):* During the time interval *T*, the rate of formation of every metabolite m is (almost) equal to its rate of degradation, i.e., all metabolite pool sizes (concentrations) remain (almost) constant during the time interval *T*. Put another way, production rate of each metabolite is equal to its consumption rate.

b. *Metabolite Pool Label Invariability (MPLI):* During the time interval *T*, all metabolite pool labels stay the same. That is, if the label of a metabolite pool is *Available*, it stays *Available* during the time interval *T*.

The PCRE property at steady-state is a natural property, referring to the state of constancy or the homeostasis (equilibrium) of the organism. As an example, in the fed state of, say, humans, *glucose*, through *Glycolysis*, is catabolized to *Acetyl CoA*, which is converted to *fatty acids* or oxidized in the *TCA Cycle*. Although *Acetyl CoA* is available to both metabolic pathways (i.e., *Fatty Acid Synthesis* and the *TCA Cycle*), it does not accumulate, as the combined consumption rate of *Acetyl CoA* by *Fatty Acid Synthesis* and the *TCA Cycle* is (almost) the same as its production by *Glycolysis*.

We use the MPLI property in order to capture a snapshot of the metabolism when metabolite pool size labels also stay constant during steady-state. Next we define some terminology.

Def'n (*Metabolic Network*). A metabolic network is a connected graph G(V, E) with a vertex set V of reactions and metabolite pools (a metabolite pool can be a substrate, regulator or product in a reaction), and a directed edge set E such that there is an edge from node u to node v if (i) v is a reaction, and u is a substrate, regulator of v, or (ii) u is a reaction, and v is a product of u.

Def'n *(ProductionRate and ConsumptionRate of metabolite pool m)*: Consider any metabolite pool m, its producer reactions p_1, p_2, …, p_i, and its consumer reactions c_1, c_2, …, c_j. Let $pr_{m,k}$ denote the *production contribution rate* of reaction p_k, $1 \le k \le i$, for metabolite m, and $cr_{m,v}$ denote the *consumption contribution rate* of reaction c_v, $1 \le v \le j$, for metabolite m, during time period *T*. Then

- $P_m = \{(p_1, pr_{m,1}), (p_2, pr_{m,2}), \ldots, (p_i, pr_{m,i})\}$ is the *active producer set* of m, where each pair $(p_i, pr_{m,i})$ refers to a producer p_i of m and its contribution rate $pr_{m,i}$; and $(pr_{m,1} + pr_{m,2} + \ldots + pr_{m,i})$ is the *ProductionRate(m)* of m; and
- $C_m = \{(c_1, cr_{m,1}), (c_2, cr_{m,2}), \ldots, (c_j, cr_{m,j})\}$ is the *active consumer set* of m, where $(c_j, cr_{m,j})$ refers to an activated consumer c_j of m and its consumption rate $cr_{m,j}$; and $(cr_{m,1} + cr_{m,2} + \ldots + cr_{m,j})$ is the *ConsumptionRate(m)* of m.

Below we formally characterize the notion of (quasi-)steady-state for the metabolism.

Def'n ((*quasi-*)*steady-state for an organism during a time period*): Given an organism *Org*, its metabolites m_l, $1 \le l \le n$, and two constants ε_{ml} and *T*, the organism *Org* is said to be in *a steady-state* during the time period *T* if

(a) ProductionRate(m_l) = ConsumptionRate(m_l) ± ε_{ml} for each m_l, $1 \le l \le n$, during the time period *T*, and

(b) Label of each metabolite m_l, $1 \le l \le n$, stays the same during the time period *T*.

2.2 Metabolite Pool Label Identifiers

The purpose of metabolite pool label identifiers is to simplify the ACT set specifications for reactions and transport processes.

Def'n *(Metabolite pool label during a time period)*: Let $T_{AVAIL}(m)$, $T_{ACC}(m)$, and $T_{SAC}(m)$, $T_{AVAIL}(m) < T_{ACC}(m) < T_{SAC}(m)$, be three threshold constants for a metabolite m, stored in the database. Given the metabolite pool m, the label of m during the time period *T* is marked with one of the following five *identifiers*.

- *Unknown* (id:-1): if the metabolite pool size for m, *Size(m)*, is unknown during time period *T*.
- *Unavailable* (id: 0): the metabolite pool size for m, *Size(m)*, is less than the threshold $_{TAVAIL}(m)$ and ProductionRate(m)) $\le \varepsilon_{ml}$ during time period *T*, where) ε_{ml} is a small constant.
- *Available* (id: 1): the metabolite pool size for m, *Size(m)*, is greater than or equal to the threshold $T_{AVAIL}(m)$ and less than the threshold $T_{ACC}(m)$ during time period *T*.
- *Accumulated* (id: 2): the metabolite pool size for m, *Size(m)*, is equal to or above the threshold $T_{ACC}(m)$, but less than the threshold $T_{SAC}(m)$ during time period *T*.
- *Severely Accumulated* (id: 3): the metabolite pool size for m, *Size(m)*, is equal to or above the threshold $T_{SAC}(m)$ during time period *T*. This label is used for the product inhibition rule BC4 of section 2.5.

There is need to use different metabolite pool labels of *Available* and *Accumulated* because, for some reactions, "availability" of a metabolite *m* as a substrate (or regulator) may be sufficient for the reaction (i) to be active through substrate availability (provided that there are no other inhibiting mechanisms) or (ii) to experience the regulating effect (i.e., inhibition/activation) of *m*, in those cases where *m* is a regulator. However, for activation/regulation, other reactions may require the "accumulation" of *m*--at least, at moderate levels. We give an example.

Example 2.1. *Acetyl CoA* is an allosteric activator of the first (also the *committed*) *step* in *Gluconeogenesis*, which is catalyzed by *pyruvate carboxylase*. And, *pyruvate carboxylase* activation needs *Acetyl CoA* accumulation. In the fed state of organism, *Acetyl CoA* is produced by *Glycolysis* (hence, is *Available*), but does not accumulate (hence has "*Not Accumulated"*). Thus, *pyruvate carboxylase* is not activated, which leads to the inactivation of *Gluconeogenesis* pathway. But, in the *fasting* state of the organism, *Acetyl CoA* is produced by *Beta Oxidation*, and consumed by the *TCA Cycle* and *Ketone Body Synthesis*. In this case, accumulation of *Acetyl CoA* occurs (slowly, but steadily), since its production rate by *Beta Oxidation* is higher than its combined consumption rate by the *TCA Cycle* and *Ketone Body Synthesis*.

2.3 Metabolite Label Condition Characterization

The metabolite label condition C about the label identifier q of a metabolite pool m is denoted as C <q, m>.

Example 2.2. Ketone Body Synthesis requires the accumulation of Acetyl CoA to use it as a substrate. Then, the required condition can be stated as C<*Accumulated*, Acetyl CoA> or, equivalently, as C<2, Acetyl CoA> when the identifier of *Available* is used.

We employ three-valued logic (*True*, *False*, *Unknown*) in evaluating conditions about metabolite pool labels of reactions.

Def'n *(Satisfaction of a metabolite label condition)*: A metabolite label condition C<q, m> is

(i) *True* if m is marked with the identifier q_{Actual} where either (a) $0 < q.id \le q_{Actual}.id$ or (b) $q.id = q_{Actual}.id = 0$ holds,

(ii) *False* if m is marked with the identifier q_{Actual} where either ($q_{Actual}.id \ne -1$ and $q_{Actual}.id < q.id$) or ($q.id = 0$ and $q_{Actual}.id > 0$),

(iii) *Unknown* if m is marked with the identifier q_{Actual} where q_{Actual}.id = -1.

Example 2.3. The condition C<*Accumulated*, Acetyl CoA> (or, C<2, Acetyl CoA>) from Example 2.2 is *True* when the corresponding pool of Acetyl CoA has the label *Accumulated* (id: 2) or *Severely Accumulated* (id: 3).

Def'n *(Negation of a Condition)*: Negation of a condition C<q, m> is denoted as ¬C<q, m>. ¬C<q, m> is *True* if m is marked with a identifier q_{Actual} such that either (a) q_{Actual}.id≠-1 and q_{Actual}.id < q.id, or (b) q.id = 0 and q_{Actual}.id > 0.

Example 2.4. The negation of the condition from Example 2.2, i.e., ¬ C<*Accumulated*, Acetyl CoA>, is *True* only when Acetyl CoA is marked as *Available* (id: 1) or *Unavailable* (id: 0) (i.e., no active producer).

Def'n *(Conflicting Conditions)*: Two conditions C_1<q_1, m> and C_2<q_2, m> which are defined on the same metabolite *m* are *in conflict* if there is no possible pool label identifier for m that would *satisfy* both C_1 and C_2.

Example 2.5. ¬C_1<*Available*, Acetyl CoA> is in conflict with C_2<*Accumulated*, Acetyl CoA>.

Def'n *(Condition Subsumption)*: Condition C_1<q_1, m> *subsumes* another condition C_2<q_2, m> if C_2 is satisfied whenever C_1 is satisfied.

Example 2.6. C_1<*Accumulated*, Acetyl CoA> subsumes C_2<*Available*, Acetyl CoA>.

2.4 Trigger Values and Activation Condition Sets

The label of a reaction r, a transport process T, or a pathway can be one of *active, inactive*, or *unknown*, as discussed next.

2.4.1. Reaction

We start with the notion of a "metabolite trigger value" for a reaction, which can be either *Available* or *Accumulated*.

Def'n *(Trigger value t of metabolite m for reaction r to be active)*: Let m be a metabolite involved in a reaction r. For r to be active, metabolite m is said to have a trigger value $t_{m,r}$, $t_{m,r} \in$ {*Available, Accumulated*}, if

(i) m is a substrate, cofactor-in, or an activator of r, and the metabolite pool identifier for m is $t_{m,r}$, or

(ii) m is an inhibitor of r, and the metabolite pool identifier for m is below (the integer id value of) $t_{m,r}$.

Each reaction r (or pathway) has a set of participating metabolite pools and their predetermined trigger values, available in a database. Each reaction (or a pathway) is associated with a set of "activation conditions", which are created based on the participating metabolites and their trigger values.

Def'n *(Activation Condition Set of a Reaction/Pathway)*: Activation condition set of a reaction (or a pathway) r, denoted as ACT(r), defines the conditions for r to be active, and is constructed as follows.

- For each m in reaction r where m is a substrate/cofactor-in/activator of r with trigger value $t_{m,r}$, C<$t_{m,r}$, m> ∈ ACT(r) where $t_{m,r} \in \{1, 2\}$ (1 and 2 are ids of *Available* and *Accumulated* labels, respectively)
- For each m in r where m is an inhibitor of r with trigger value t_{mr}, ¬C<$t_{m,r}$, m> ∈ ACT(r) where $t_{m,r} \in \{1\}$
- For each m in r where m is a product/cofactor-out of r, ¬C<3, m> ∈ ACT(r) (Product Inhibition rule BC4; 3 is the id of *Severely Accumulated* label).
- If the ratio T=size(m_1)/size(m_2) of energy metabolite pairs is specified as an activator for r, then C_1(*Accumulated*, m_1)∈ACT(r), and ¬C_2(*Accumulated*, m_2)∈ACT(r). If T is an inhibitor for r, then ¬C_1(*Accumulated*, m_1)∈ACT(r), and C_2(*Accumulated*,m_2)∈ACT(r)

The activation condition set ACT of a given reaction is defined a priori (offline) before any metabolomics analysis is carried out.

2.4.2. Transport Processes

We view each transport process $T_{c1\text{-}to\text{-}c2}$ as having one metabolite transported from compartment c1 to compartment c2, subject to the activation condition set ACT for $T_{c1\text{-}to\text{-}c2}$. We give an example.

Example 2.7. The transport process $T_{bl\text{-}to\text{-}muscle}$ of *glucose* from blood to muscle may be characterized with the ACT($T_{bl\text{-}to\text{-}muscle}$ (*glucose*, blood, muscle)) as {C<*Available*, blood.glucose>, C<*Available*, blood.insulin>}. That is, for *glucose* to be transported from blood to muscle, both *glucose* and *insulin* must be at least *Available*. On the other hand, the transport $T_{muscle\text{-}to\text{-}bl}$ of glutamine from muscle to blood can be conditioned based on its availability in muscle, i.e., ACT($T_{muscle\text{-}to\text{-}bl}$ (glutamine, muscle, blood)) = {C<*Available,* blood.glutamine>}.

We have the following transport process rules.

Rule TR1. Let c1 and c2 be two compartments, m be an observed metabolite in compartment c1, and $T_{c1\text{-}to\text{-}c2}$ (m, c1, c2) be m's transport process from c1 to c2. Assume that pool label of m in c2 is *Unknown*. Then if ACT($T_{c1\text{-}to\text{-}c2}$) is satisfied then $T_{c1\text{-}to\text{-}c2}$ (m, c1, c2) is *active;* otherwise, it is *inactive*.

Rule TR2. For active transport processes (i.e., the ACT set is satisfied), we assume that the metabolite pool of the product has the same label with the substrate.

Rule TR3. For transport processes, the product inhibition rule (Please see rule BC4 of Section 2.5) does not apply.

2.4.3. Steady-State Labels for Reactions and Transport Processes

We define the *steady-state label* of a reaction/transport process as one of *active, inactive*, or *unknown*, based on the satisfaction of its associated activation condition set ACT.

Def'n *(active, inactive, or unknown reaction/transport process state)*: Given a reaction/transport process r with an associated activation condition set ACT(r) defined on the participating metabolites, r is said to be *active* (i.e., having a nonzero flux) during the steady-state time period if

(i) All conditions in ACT(r) are satisfied; i.e., all conditions that involve substrates, cofactors, and products of r are satisfied, and

(ii) Among the conditions involving regulators of r, those conditions that include regulator(s) with the highest precedence are satisfied.

Reaction/transport process r is *inactive* if there is at least one unsatisfied condition in ACT(r). Otherwise, the state of r is *unknown*.

Note that, for some reactions there may be multiple activators and inhibitors, in which case, we assume that (a) we have a priori information about the precedence of regulators, and (b) we make use of such precedence information in deciding whether the reaction is active or inactive.

2.5 Biochemistry-Based Rules

Next, we list a number of basic biochemistry (BC)-based rules that we use in the rest of the paper.

Rule BC1. For each reaction, when multiple regulators with conflicting regulatory effects (activation or inhibition) on an enzyme are in place, the regulator with the strongest effect (highest precedence) on the enzyme is considered, and the other regulators are ignored.

The regulated reactions in a pathway may be classified as *rate-limiting* and *committed* steps. Once the *committed step* takes place, other reactions in the pathway follow this reaction until the end-product is produced, provided that none of the other regulated processes are blocked or inhibited. A committed step of a pathway is usually one of the early irreversible reactions in the pathway. As an example, in glycolysis, the committed step is the same as the rate-limiting step, *PFK1*.

Rule BC2. If the committed step of a pathway p is blocked (i.e., inactive), then p is inactive (i.e., all reactions in p are inactive).

We associate each compartment with particular pools of metabolites as its input and output. We then connect two compartments in the metabolic network if a transport process connects the two.

Rule BC3. Each input and/or output metabolite of a compartment is associated with a transport process (precaptured and modeled in the database). A transport reaction and an enzymatic metabolic reaction are connected if they share at least one metabolite pool (i.e., as their substrate and/or product).

Due to similarities in the way they bind to enzymes, substrates are in competition with products to bind to their enzymes. As the concentration of products increase, this competition slows down the rate of enzymes binding the substrates. Hence, the reaction rate decreases. Eventually, when the product accumulation reaches to high levels, the corresponding reaction is inhibited dramatically.

Rule BC4. Whenever a non-bio-fluid metabolite m is marked as "*severely accumulated*", all reactions that produce (and, therefore, due to the steady-state assumption) and consume m are "*inactive*".

The next set of rules follows from the steady-state assumption.

Rule BC5. If all producers (consumers) of a metabolite pool m are inactive then, due to the PCRE property, regardless of the pool label of m, all consumers (producers) of m are inactive.

Rule BC6. If at least one producer (consumer) of a metabolite m is active, then (i) m is either available or accumulated, and (ii) at least one consumer (producer) of m is active.

Rule BC7. If the metabolite m is *Unavailable* then all consumers (and, thus, due to the steady-state assumption) and all producers of m are inactive.

Rule BC8. Substrate and product labels of a transport process with no conditions are always the same.

Next, using rules BC1-8, we specify the notion of "inconsistent" metabolite pool and reaction label assignments.

Def'n *(Inconsistency)*: For each Rule BC_i, $1 \leq i \leq 8$, violation of Rule BC_i in terms of metabolite pool and/or reaction label assignments constitutes an inconsistency in metabolite pool and reaction labels.

For example, as a product of an *active* reaction r, the label of metabolite pool m should not be *Severely Accumulated*, since it violates Rule BC4.

3 ACTIVE/INACTIVE GRAPH GENERATION, EXPANSION-MERGE

Starting from a given set of observations, SMDA employs iterative backward and forward reasoning with the goal of identifying possible metabolic mechanisms which may have led to the observed changes. Please see the supplement [8] for the details of the SMDA expansion and merge algorithm.

4 COMPUTATIONAL PERFORMANCE EVALUATION

In this section, SMDA algorithm is empirically evaluated, and different expansion strategies are compared with real data.

4.1 Experimental Settings

Environment. The experiments are performed on a Dell PowerEdge R710 Server with two Intel® Xeon® quad processors and 48 GB main memory, running the Windows Server 2008. The web application server is Microsoft IIS 7. The database server is Microsoft SQL Server 2010. The SMDA web site is implemented with Microsoft ASP.NET; and the client visualization is implemented with Java.

Database. The metabolic network database, constructed from data in the literature and continually expanded, includes mammalian metabolic pathways that are built for PathCase Metabolomics Analysis Workbench, with 22 pathways, 202 metabolites, 375 metabolite pools, and 240 reactions. The thresholds are set up according to the Human Metabolome Database [10].

Observations. Metabolomics observations used in experiments are from cystic fibrosis mice metabolomics profiles.

4.2 Experimental Results

A. *Relationship between the number of observations and the number of G_{AI} and R-graphs.*

In this experiment, we evaluate the performance of SMDA for different number of user observations. We experiment with three different size sub-networks. For each sub-network, we change the number of metabolite pool observations and record the number of graphs in the result, as listed in Table 4.1.

Observation 1. *For small sub-networks, a linear increase in the number of observations results in an exponential decrease in the number of G_{AI} and R-graphs in the output.*

From Table 4.1, regardless of the size of the sub-network, the number of G_{AI} and R-graphs decreases as we provide more observations as input. Note that, in some cases, increasing the number of observations will not reduce the number of graphs, since there is only one possible label for the input pools in the results. Then the input pool observation is really duplicate information with no reduction on the result size.

Sub-Network	# Reactions	# M. Pools	# Observations	# G_{AI}-graphs	# R-graphs
Pentose pathway	8	16	1	8938	846
			2	860	423
			3	588	376
Glycolysis pathway	14	25	1	152	12
			2	8	8
			3	4	4
Glycolysis+TCA Cycle pathways	24	48	2	332288	160
			4	166144	80
			6	128	32

Table 4.1. The number of observations versus the number of output graphs for small sub-networks.

In another experiment, for a larger sub-network, we observe how the algorithm scales. We choose a connected sub-network with 6 pathways, 48 reactions and 132 metabolite pools. The number of G_{AI} and R-graphs versus different numbers of observations is shown in Table 4.2.

# Reactions	# M. Pools	# Observations	# G_{AI}-graphs	# R-graphs
48	132	17	3072	40
		23	1536	20
		31	384	12
		33	192	12
		35	192	12
		37	192	12

Table 4.2. The number of observations versus the number of graphs for a large network.

From Table 4.2, we can see that, even in a large sub-network, we can get reasonably small numbers *of* G_{AI} *and* R-graphs with increased number of pool observations.

Observation 2. *For larger sub-networks, a linear increase in the number of observations results in an exponential decrease in the number of G_{AI}- graphs and a linear decrease in the number of R-graphs in the output.*

B. *Algorithm time efficiency.*

The execution time is composed of two parts: expansion time and merge time. For each sub-network, we execute each of the three expansion strategies. The results show that, in general, increasing the no. observed pool observations decreases the execution time exponentially. This is due to the fact that, with more observed values, expansion time is decreased exponentially by reducing the expansions of many small sub-networks, instead of one large network. However, in some experiments, increasing the number of pool observations has actually increased the execution time, instead of decreasing it. In those cases, we have found that merge time costs are significantly higher than expansion time costs.

Observation 3. *A linear increase in the number of metabolite pool observations results in an exponential decrease in the execution time of the algorithm.*

Figure 4.1 shows how the algorithm behaves with "Selective expasion1" strategy. The results are similar for "Naïve expansion" and "Selective expansion2" strategies.

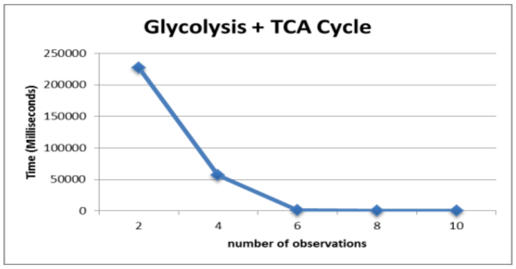

Figure 4.1. SMDA time cost for a single network versus the number of observations for Glycolysis and TCA Cycle combined.

C. *Comparing expansion strategies for a large sub-network.*

Next we use the connected sub-network of Table 4.2 with 6 pathways, 48 reactions and 132 metabolite pools. Figure 4.2 shows execution times of different expansion strategies.

SMDA employs [7] three different expansion strategies during the G_{AI} graph expansion stage, namely, the na ïve expansion, the selective expansions #1 and #2. Since "Selective Expansion #1" excludes the set of energy metabolite pools during the expansion, it takes less time than other two expansion strategies when the observations are less.

Observation 4. *Selective Expansion#1 time costs are invariably much less than the time costs of Na ïve Expansion and Selective Expansion#2 Strategies.*

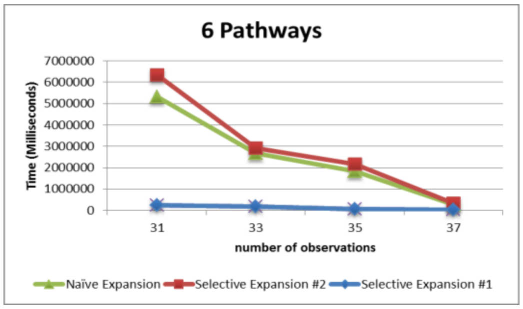

Figure 4.2. Expansion strategy times for a sub-network with 6 pathways

5 CONCLUSIONS AND FUTURE WORK

SMDA is currently being evaluated extensively for (i) its usefulness in a cystic fibrosis research, and (ii) in reproducing similar results (when constrained to 0/1 flux values) to other metabolomics-related in silico studies that characterize all allowable states. Other future research directions include (a) incorporating *Exploratory Data Mining and Knowledge Discovery* capabilities for the SMDA query output search space, and (b) adding more precision to its selected/discovered "hypothesis" (i.e., an activation/inactivation scenario) by estimating flux rate ranges via the use of constrained-based techniques [2, 3, 4].

6 ACKNOWLEDGMENTS

This research is supported by the National Science Foundation under grants DBI-0849956 and DBI-0743705, and the National Institute of Health under grant R01 GM088823.

7 REFERENCES

[1] Yizhak, K. et al, *Integrating quantitative proteomics and metabolomics with a genome-scale metabolic network model*, Bioinformatics, Vol. 26, ISMB 2010, pp 1255-1260.

[2] Price, N.D., Reed, J.L., Palsson, B.O., *genome-scale models of microbial cells: Evaluating the consequences of constraints.* Nature Reviews, Vol. 2, Nov. 2004, pp. 886-897.

[3] Joyce, A.R., Palsson, B.O. *Towards whole cell modeling and simulation: comprehensive functional genomics through the constraint-based approach.* Progress in drug research, Vol. 64, 2007.

[4] Oberhardt, M.A., Palsson, B.O., Papin, J.A. *Applications of genome-scale metabolic reconstructions.* Molecular Systems Biology, 5:320, 2009.

[5] Schuster, S., Fell, D.A., Dandekar, T. *A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks.* Nature Biotechnology Vol 18, March 2000, pp. 326-332.

[6] SMDA tool, available at http://nashua.case.edu/pathwayssmda/web

[7] PathCase family of applications, available at http://nashua.case.edu/pathwaysweb

[8] Supplement to this paper, available at http://nashua.case.edu/pathwayssmda/smda.supplement.pdf

[9] Cakmak A., Dsouza A., Hanson R. W., Ozsoyoglu Z. M. 2010. *Analyzing Metabolomics Data for Automated Prediction of Underlying Biological Mechanisms*. ACM Bioinformatics and Computational Biology Conference, Niagara Falls, NY.

[10] Human Metabolome Database, available at http://www.hmdb.ca/biofluid_browse