# SWARM: A Scientific Workflow for Supporting Bayesian Approaches to Improve Metabolic Models

Xinghua Shi[1]
[1]Department of Computer Science
University of Chicago
Chicago, IL 60637, USA
shi@uchicago.edu

Rick Stevens[1, 2]
[2]Mathematics and Computer Science
Argonne National Laboratory
Argonne, IL 60439, USA
stevens@anl.gov

## ABSTRACT
With the exponential growth of complete genome sequences, the analysis of these sequences is becoming a powerful approach to build genome-scale metabolic models. These models can be used to study molecular components, their activities and relationships, and thus achieve the goal of studying cells as systems. However, constructing genome-scale metabolic models manually is time-consuming and labor-intensive. This property of manual model-building process causes the fact that much fewer genome-scale metabolic models are available comparing to hundreds of genome sequences available. To tackle this problem, we design SWARM, a scientific workflow that can be utilized to improve genome-scale metabolic models in high-throughput fashion. SWARM deals with a range of issues including the integration of data across distributed resources, data format conversions, data update, and data provenance. Putting altogether, SWARM streamlines the whole modeling process that includes extracting data from various resources, deriving training datasets to train a set of predictors and applying Bayesian techniques to assemble the predictors, inferring on the ensemble of predictors to insert missing data, and eventually improving draft metabolic networks automatically. By the enhancement of metabolic model construction, SWARM enables scientists to generate many genome-scale metabolic models within a short period of time and with less effort. The availability of a large number of metabolic models will lead to a new generation of important biological hypotheses and experimental designs based on the analysis of these models.

## Categories and Subject Descriptors
J.3.1 [**Computer Applications**]: Life and Medical Sciences – *Biology and genetics.*

## General Terms
Design, Management.

## Keywords
Scientific workflow, Bayesian approaches, Metabolic models.



*CLADE'08*, June 23, 2008, Boston, Massachusetts, USA.



## 1. INTRODUCTION
High-throughput sequencing technology in biology and automated genome annotation tools in bioinformatics make it possible to identify and assign functions to most metabolic genes in an organism. These gene functions then can be mapped to biochemical reactions that characterize the capabilities of genes when they are activated. The sequence and biochemical information, together with the strain-specific information of an organism, can be integrated and assembled to build a genome-scale metabolic model. A genome-scale metabolic model attempts to capture and represent "all" that is known about the organism from annotated genome sequence. Such a model can be used to study how an organism performs under various conditions and what systemic properties the network possesses.

Extensive research has been conducted on the methods that are applied to build metabolic models. Due to the lack of detailed kinetic information, a constraint-based approach, flux balance analysis (FBA) [1-3], has been proposed to assess theoretical capabilities and operative models of metabolic networks. FBA can be applied to study genotype-phenotype relations, identify essentiality of genes, and investigate different states the cell may have under different situations and so on. At present, FBA is the only methodology by which genome-scale models have been constructed. [1]

Although FBA has been under research over 20 years and approximately 18 genome-scale metabolic networks for 14 organisms and cells [4-18] have been built, the speed of constructing metabolic networks cannot catch up with the growth of the number of organisms with annotated genomes. The number of complete genomes is more than 400 hundred and it is expanding to reach 1000. This big gap between the number of genome-scale metabolic models and the number of available complete genomes is largely due to the fact that most of these models are reconstructed manually. Building a genome-scale metabolic model with thousands of metabolites and reactions is time-consuming and labor-intensive. With the number of annotated genomes expanding to thousands, it is desirable that we produce complex metabolic models in a high-throughput manner.

To address this problem, we propose and design a scientific workflow for supporting Bayesian approaches to improve metabolic models (SWARM). SWARM deals with a range of issues including the integration of data across different resources, data format conversions, data update, and data provenance. Putting altogether, SWARM streamlines the whole model building procedure by automating the following processes: extracting data from various resources; deriving training datasets

to train a set of predictors and use Bayesian techniques to assemble these predictors; inferring on the ensemble of predictors to insert appropriate data; and eventually improving draft metabolic networks in a high-throughput way. By the enhancement of automated metabolic model construction, SWARM enables scientists to generate thousands of genome-scale metabolic models within shorter period of time and with less effort.

The remainder of this paper is organized as follows. Section 2 introduces the background of scientific workflows and metabolic modeling, and reviews some related work afterwards. Section 3 discusses the problem of improving the construction of genome-scale metabolic models automatically and explains the motivations of our work. Section 4 describes our design and essential elements of the SWARM workflow. Section 5 explains the validation of SWARM and a brief summary with discussions is given in Section 6.

# 2. BACKGROUND AND RELATED WORK

Building a scientific workflow is desirable to facilitate the development of metabolic models in the rigorous bioinformatics and systems biology areas. In this section, we review the background of scientific workflows, with a focus on bioinformatics workflows. Then we discuss the research that has been done in developing metabolic models and present related tools.

## 2.1 Scientific Workflows

Scientific workflows attempt to automate scientific processes in which tasks are structured based on their control and data dependencies [40]. These workflows facilitate scientists to build and validate models automatically or semi-automatically, by taking a series of steps to collect, analyze, execute, process, debug, manage, and visualize data. The goal of building scientific workflows is to better support scientists to do their research and promote e-Science.

Lots of efforts have been made to build scientific workflows and scientific workflow systems, particularly in the Grid [39] community. Yu and Buyya [40] presented a taxonomy that characterizes and classifies various approaches for building and executing workflows on Grids. In [41], Barker and Hemert reviewed the existing business and scientific workflows and presented key suggestions towards the future development of scientific workflow systems.

There are a number of scientific workflow systems that have been proposed and designed. We list some of the scientific workflow systems that can be applied to bioinformatics and life sciences. Kepler [42] is an open-source scientific workflow system that aims to simplify the access and process of scientific data in various domains, with support of web service-based workflows and Grid extensions. Kepler provides a formal model for scientific workflows based on an actor-oriented design [43] and introduces a hybrid type system that separate structural data type from semantic type. Taverna [44] is an open-source, Grid-aware workflow system that constructs and executes workflows for the lift science community. As a part of the myGrid [45] project, Taverna enables the scientists to describe and execute their experiment processes in a structured, repeatable and verifiable way. GPFlow [46] provides a scientific workflow environment that supports bioinformatics experiments by wrapping legacy tools and presenting an interactive web-based interface to scientists. ASSIST [47] is a programming environment that allows the design of bioinformatics workflows that can be executed on Grid. Swift [48] is a workflow system that supports the specification, execution, and management of large-scale workflows on Grid.

However, the problem we try to solve needs a new set of components whose specifications cannot be defined in existing workflow systems [42-48]. Moreover, the existing workflow systems don't support mechanisms to experiment with various approaches for learning and modeling. With the exploration of designing a domain-specific workflow to improve metabolic modeling, it is possible then to generalize workflow components and fit them into existing workflow systems.

## 2.2 Metabolic Modeling

Metabolic Modeling has a long history of research and important impact on biology. Generally speaking, there are two primary approaches to build metabolic models: dynamic and static modeling. The dynamic modeling method intends to simulate cellular processes based on fundamental physicochemical laws and principles. Although dynamic modeling can produce a detailed look at metabolic networks, it requires kinetics information that might not be available and ensues huge computational cost. Due to the lack of quantitative kinetics data and detailed information about every enzyme and cofactor, an alternate modeling approach, static modeling, such as flux balance analysis (FBA) [1-3] was proposed. The FBA approach views the metabolism as a continuous process and studies the steady status of this process. Hence, FBA is based on the steady-state hypothesis that at the time of study, the network is at steady state and each metabolite is balanced even though there are fluxes in and out of this metabolite. This steady-state assumption is valid for metabolic networks because metabolic transients are much faster compared to both cellular growth rates and dynamic changes of the environment.

Building a FBA model only requires information about metabolic reaction stoichiometry, medium that the organism may grow on, and the measurement of a few other organism-specific parameters. All of this information defines the domain of allowable flux distributions that may be taken to define an organism's metabolic phenotypes. Within this allowable domain, a single optimal flux distribution is sought based on assumed objective function, with the aid of Linear Programming techniques. As an approach to model an organism's systemic behavior and make quantitative predictions with the absence of detailed kinetics, FBA is feasible to build genome-scale metabolic models. These genome-scale FBA models have a wide range of applications. They can be used to interpret metabolic network behavior, study metabolic states and analyze the capabilities of a metabolic network, manipulate a metabolic network to produce certain desired products, and generate quantitative hypotheses in silico that may be tested by wetlab experiments[3,6].

Over the past 20 years after FBA was firstly proposed, this approach has been studied extensively to construct genome-scale metabolic networks. To our best knowledge, approximately 18 genome-scale metabolic models for 14 organisms and cells have been built based on FBA approach. These models have been proven successful in performing whole-cell studies with explanatory and predictive capacity. Even in cases where FBA

fails to explain experimental data, the formal treatment and analysis of a metabolic network provide powerful tools for representing and refining knowledge [3,6].

## 2.3 Related Tools

One important aspect of building a genome-scale metabolic model is to fill the network holes inside the network. These network holes happen when the model is incomplete due to missing data. Missing data can be anything from missing genes, non-annotated genes, to proteins with no reactions associated in databases. In order to fill network holes, different types of efforts from various groups are carried out.

Osterman and Overbeek [19] proposed to accelerate the pace of discovering missing genes by comparative analysis of a large and growing number of diverse sequenced genomes. The SEED project [20, 21] is such a peer-to-peer environment to enable distributed teams of researchers to rapidly annotate genomes, especially microbial genomes. By providing a set of open-source comparative genome annotation and analysis tools, SEED enables researchers to create, collect, and maintain sets of gene annotations organized by groups of related biological and biochemical function roles across many organisms. These groups of related function roles are called subsystems, and each subsystem is essentially a set of biological functions that together implement a specific process. Function roles are then mapped to biochemical reactions as those accumulated in KEGG [22,23]. Kharchenko *et al.* [24] presented a computational approach for identifying genes encoding missing metabolic enzymes in a partially reconstructed metabolic network using coexpression properties of the metabolic network. By extending this method, in [25], Kharchenko *et al.* provided a mechanism to identify genes encoding for a specific metabolic function based on local structure of metabolic network and multiple types of functional association evidence, including clustering of genes on the chromosome, similarity of phylogenetic profiles, gene expression, protein fusion events and others.

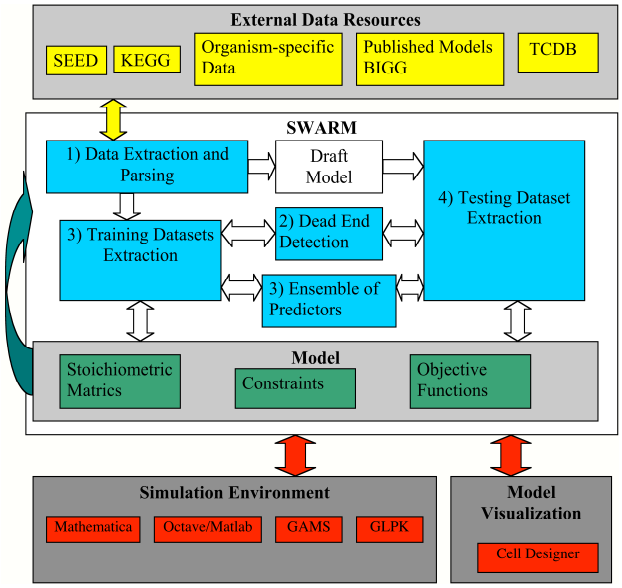


**Figure 1. Overview of Modeling Workflow in SWARM**

Beyond all the manually built models we discussed in Subsection B, certain efforts have been carried out to build models in an automatic fashion. In [26], DeJongh *et al.* presented their mechanisms in the generation of substantially complete metabolic networks for over 400 complete genome sequences currently in SEED. Their tools extend subsystems in the SEED to represent reaction subnetworks, enhance the curation of associations between functional roles and reactions in subsystems, assemble and verify reaction subnetworks in subsystems. These efforts enable better gene-protein-reaction associations and provide better genome-scale metabolic network reconstruction out of SEED. But this approach introduces reactions without showing background evidences and more importantly, the reconstructed metabolic networks are still incomplete and often contain network holes. In order to generate valid models and explain the reasons of inserting some reactions, further investigation of methods to fill in network holes is needed.

Becker *et al.* [27] presented CORBA, a toolbox running in the Matlab environment, which allows quantitative prediction of cellular behavior using FBA approach. Specifically, this software allows predictive computations of both steady-state and dynamic optimal growth behavior, effects of gene deletions, comprehensive robustness analyses, sampling the range of possible cellular metabolic states and determination of network modules. SimPheny [28] is a commercial software platform that enables efficient development of genome-scale metabolic models of microbial organisms and their simulation using a constraint-based modeling approach like FBA. FluxAnalyzer [29] is a package for MATLAB to explore structure, pathways, and flux distributions in metabolic networks. CellNetAnalyzer [30] is the successor of FluxAnalyzer and can be utilized to analyze the structure and function of signaling and regulatory networks. However, to our best knowledge, these tools don't provide a mechanism to analyze network connectivity and introduce plausible reactions to fill in network holes. PathoLogic [31,32] is a set of software that uses Bayesian methods to identify missing enzymes in predicted metabolic pathway databases. Their system is mostly focused on inferring on individual pathways, while filling network holes in a genome-scale is still an issue.

# 3. THE PROBLEM AND MOTIVATIONS

In this section, we elaborate the problem of constructing scientific workflows to extend genome-scale metabolic models for available complete genomes, and present motivations of our work.

Currently, approximately 18 genome-scale metabolic models of 14 organisms and cells have been built. But there are hundreds of complete genome sequences in databases, for example, there are 505 complete genomes and 476 complete bacteria genomes in the SEED. With the rapid growth of complete genomes, Overbeek *et al.* [33] expects the SEED system to support rapid annotation of the first 1,000 genomes to be sequenced. With thousands of annotated genomes to be available, it is demanding that corresponding genome-scale metabolic models be generated as well. In this scenario, producing so many large-scale models by hand would be an extremely labor-intensive and time-consuming task. A more desirable solution is to build a workflow that can automate the model building process including the following iterative steps as illustrated in Figure 1:

1) Extract data from external data resources including the KEGG, SEED, BIGG, TCDB, and other data resources;

Parse and convert different data formats into a congruent internal data format;

2) Analyze the connectivity of draft metabolic networks gathered from the SEED; Detect dead end metabolites where network holes appear;

3) Derive training datasets from the data collected, design a set of predictors to take account of network properties and biological evidences, train predictors on training sets, and apply methods such as Bayesian approaches to integrate individual predictors;

4) Derive testing datasets and use the ensemble of predictors trained in step 3) to infer plausible reactions that could be inserted into draft incomplete metabolic networks and other test networks; A collection of reactions are selected to insert into networks, according to the results of predictors;

5) Generate FBA models including stoichiometric matrices, constraints and objective functions; Use simulators with Linear Programming package such as Mathematica, Octave/Matlab, GLPK, and GAMS to run simulations; If the models are valid, using the simulation results to predict what these metabolic models could produce under various conditions and validate the properties of models; These valid models can also be visualized using graphviz [34] or Cell Designer [35]; If no valid model is generated, go back to previous steps and debug models iteratively.

It is more often that it may take many loops of these steps to generate a valid metabolic model. Hence the process to construct a genome-scale model requires a large volume of repetitive work and continuous tries. In particular, lots of efforts are needed to select reactions when there are network holes, which occur due to the lack of certain information from data resources. In order to fill in these network holes, it is essential to study and debug the topology of the metabolic networks. The problem of debugging network connectivity and fixing network holes is an essential issue to construct genome-scale metabolic models. Most available FBA models are created with much work and a large amount of time dedicated to fix networks holes. As time goes, the problem of manual model building process is more prominent, with thousands of annotated genome sequences available. In order to build a genome-scale metabolic model for each organism that has been and to be sequenced and annotated, it is desirable to automate this model building process.

However, it is a challenge to automate the construction of genome-scale metabolic models, thanks to the complexity of metabolic networks and the searching for appropriate data to fix network holes. Faced up with this challenge, we propose SWARM, a scientific workflow that addresses individual problems and integrates the model building process to expand genome-scale metabolic models automatically. By producing genome-scale metabolic networks in mass-production way, SWARM will accelerate the process of improving metabolic models, based on incomplete knowledge available.

The modeling workflow as discussed in Figure 1 can be viewed as a front-end workflow that deals with modeling. To make it possible, there is the other back-end scientific workflow that we develop in the aim of supporting this front-end modeling workflow. There are many issues that need to be addressed in designing and developing the back-end scientific workflow of SWARM. We present a list of imperative issues as following:

**Table 2. Example of Gene Identifiers in Various Databases**

| SEED | KEGG | BIGG |
|---|---|---|
| figl83333.1.peg.2111 | eco:b2137 | b2137, yohF |
| figl83333.1.peg.4176 | eco:b4266 | b4266, idnO |

**Table 1. Example of Reactions in Various Databases**

| KEGG | BIGG |
|---|---|
| R00226 | ACLS |
| R02142 | XPPT |

1) *Version control problem*: The databases we extract data from, especially SEED and KEGG, are active and frequently update their data. Therefore, this updated information needs to be piped into SWARM, and reflected in the computational and modeling process of the workflow. It should be possible to record data provenance in SWARM and keep models up to date.

2) *Integration, representation and reconciliation of data from various resources*: In bioinformatics and systems biology, it is an important issue to map data from various resources to a common name space. Although efforts have been carried out to unite different data name spaces into an integrated format such as the work of SBML [36] and Gene Ontology [37], there are still a large volume of legacy and upcoming data with distinct and even incompatible formats. In this situation, integration, representation and reconciliation of data from different resources by parsing and mapping is still an essential process. A simple example would be the mapping of gene identifiers across different databases such as in SEED, KEGG and BIGG. As shown in Table 1, the same gene could have distinct identifiers in different databases. Table 2 shows different reaction names and formulas in KEGG and BIGG.

3) *Exception handling*: A unique property of biological systems is that there are all kinds of exceptions. Although a majority of data name spaces can be converted to each other, there are exceptions and a collection of data name spaces are hard or even cannot be mapped to others. For example, there is no matching KEGG reaction for BIGG reaction UNK3.

Therefore, in order to build a scientific workflow that supports the extension of metabolic models automatically, it is required to solve all of the problems mentioned above.

# 4. SYSTEM OVERVIEW AND ESSENTAIL ELEMENTS OF SWARM

In this section, we overview the system design of SWARM and elaborate essential parts of the workflow that deals with representation and reconciliation of data from various resources, computational complexity and challenges in streamlining the process of extending metabolic models using data available. We first present the infrastructure of SWARM, a scientific workflow to improve genome-scale metabolic models automatically, and describe the workflow of using SWARM to build metabolic models. We then explain essential elements of SWARM in detail.

## 4.1 System Overview

The process of building metabolic models based on FBA approach is an iterative procedure that starts with extracting stoichiometric information from genome annotations. The ultimate goal of these metabolic models is to predict phenotypes under certain conditions, given the fluxes generated by FBA approach. In this scenario, we propose SWARM, a scientific workflow that automates the extension of metabolic models using Bayesian techniques.

As illustrated in Figure 1, SWARM takes input from external databases to collect information about all known reactions, compounds, spontaneous reactions, organism-specific data including genome annotations, proteins those genes encode for, reactions those proteins catalyze, transporters those transport proteins catalyze, and biomass composites of the organism. Afterwards, all of the information is parsed and split into training datasets and testing datasets in SWARM. We retrieve SEED and KEGG for gene annotations, reactions, compounds, subsystems and pathways.

The core infrastructure of SWARM contains three main components: the parser that translates data with different name spaces from external databases to the unique data name space used in SWARM; deadend detectors that check connectivity of a draft metabolic network and search for deadend metabolites where network holes occur; the ensemble of predictors which is trained on training dataset and used to run on testing data. The assembled predictors generate candidate reaction lists for draft models, based on various evidences including network topology, gene co-occurrence profiles, gene clusters on chromosomes, KEGG Orthology, KEGG pathway maps and network modules. After filling network holes with selected reactions from candidate lists, metabolic models are fed into simulators with various linear programming environments. Based on simulation results, properties of a metabolic model can be validated and verified. These properties include that the cell can grow, interactions among metabolites should agree with biochemistry and the cell should possess certain extent of robustness. These models can also be visualized using network visualization tools like graphviz or CellDesigner.

A bigger picture of the Chicago Systems Biology Global Workflow is shown in Figure 2. SWARM sits right after (divided by a dotted vertical line) SEED and contributes to the global workflow significantly by completing it. SEED is tended to accumulate complete genomes, perform semi-automated feature identification and annotation, run subsystem analysis and finally determine a reaction set that comprises a draft model. Afterwards, SWARM generates stoichiometric matrix from the reaction set, insert spontaneous reactions and transport information to build a FBA model. Based on constraints and observations of the model, we can predict phenotypes under these constraints and observations. Combing SEED and SWARM together, the global workflow predict genotype-phenotype relationships and carry out systemic analysis on thousands of complete genomes.

## 4.2 Essential Elements of the Workflow

SWARM takes input from various distributed and local data resources, processes them computationally and outputs genome-scale metabolic models. Five essential elements of SWARM can be categorized as the following.

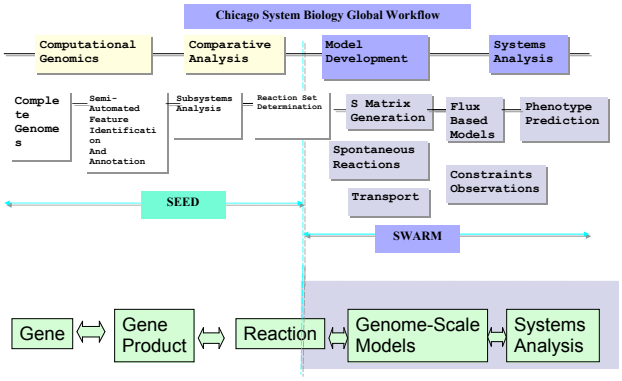


**Figure 2. The Chicago System Biology Global Workflow**

1) Integration, representation and reconciliation of various data:

In bioinformatics, it is of great importance to integrate data from various resources, cross-reference and match them, discover related piece of information such as aliases, reconcile unmatched or incompatible information. The data collected as inputs to SWARM has different and even incompatible name spaces. In general, data used in SWARM is extracted from three main distributed resources: KEGG, SEED and existing genome-scale metabolic models including BIGG, which is a repository of genome-scale metabolic models. Transport information extracted from TCDB and published models, together with a set of spontaneous reactions is accumulated as well. Since data from these different places have different name spaces, it is required to convert them into a common name space in SWARM.

Faced up with this problem, we study dependencies of these data name spaces and build a set of tools to streamline the mapping process which project different name spaces to a common name space. The first category of mapping is focused on building a mapping table inside individual data resources. Since we extract annotated sequences from SEED and based our workflow on the SEED environment, we firstly extract mappings among SEED gene identifiers, gene aliases (the genes names in other databases including in KEGG), functional roles, EC numbers, FigFam (protein family in SEED) identifiers, and KEGG reaction identifiers. Secondly, from KEGG, we extract mappings among KEGG gene names, K numbers (KEGG Orthology, abbreviated as KO), and KEGG reaction identifiers. We also download other information from KEGG including the organism list, compound list, reaction list, KO list, reaction lists, pathway maps and network modules. Thirdly, from published genome-scale metabolic models including BIGG, we extract mappings between gene abbreviations and reaction identifiers used in existing models. For each model, files containing gene-protein-reaction associations, compound and reaction lists, biomass compositions and other information are extracted as well. These files related to existing models are generated by our mapping tools, running on the data in the supplementary of their papers or that stored in BIGG. Notice that BIGG uses a distinct set of representations of genes, compounds and reactions.

Besides building mapping tables inside individual data resources, in order to streamline the workflow, we generate a large collection of tables mapping across different resources. These tables include

mappings from SEED organism names to KEGG organism abbreviations, from SEED gene identifiers to KEGG gene identifiers, from KEGG genes to SEED protein families (FigFams), from KO identifiers to Figfams, from BIGG genes to SEED genes, from BIGG genes to FigFams, from KEGG compound identifiers to compound abbreviations used in BIGG, from KEGG reactions to BIGG reactions.

Most tables are built automatically by tools, nonetheless, manual work is involved in mapping KEGG compound to BIGG compound and KEGG reaction to BIGG reaction. In the process of reconstructing Staphylococcus aureus N315 [7] and Escherichia coli K-12 [5] to use them as our reference models, we extract compound lists in the two published models. For compounds in the two models that don't have corresponding KEGG compound identifiers, we look at their chemical formulas and try to expand the mapping table to include compound mappings that have reasonably similar chemical formulas.

As a result of the work in [26] to reconstruct existing model of Staphylococcus aureus N315, a list of mapping KEGG reactions to BIGG reactions is produced. The authors also curated the mapping of SEED function roles to KEGG reactions both manually and automatically by incorporating their scenario mechanism into SEED. From the table of KEGG reactions to BIGG reactions generated by [26], we keep a growing list of the mapping from KEGG reactions to BIGG reactions. For example, the E.coli iJR904 model has 931 reactions including 747 reactions and 184 transports. Out of the 747 reactions, 515 BIGG reactions were mapped to KEGG ones by [26]. 232 reactions plus 184 transports in the iJR904 model are not mapped to KEGG reactions.

There are no transport reactions in KEGG, so we write tools to construct transports based on the mapping of BIGG compounds and corresponding KEGG compounds. This generates 180 transport reactions in SWARM out of 184 transports in the iJR904 model. As shown in Table 3, T00021, where E02917 is an extracellular compound and C02917 is an intracellular compound, is generated for BIGG transport 12PPDt, where '[e]' indicates extracellular compound and '[c]' stands for intracellular compound, based on BIGG compound "12ppd-S" is mapped to "C02917". Only 4 BIGG transports cannot be created since there are no matching KEGG compounds.

As a fact, capturing transport information is a very difficult issue to extend metabolic models. Transports are those important fluxes that carry specific nutrients, ions, etc. through cell membranes. Unfortunately, they are not well recorded or annotated. In order to characterize fluxes in and out of the cell membrane of an organism, a complete list of transports for an organism is needed. Although KEGG incorporates a growing database of thousands of biochemical reactions, it does not include transport information so far. SEED is in the process of incorporating more transport information into annotations of genomes, but there are still a limited number of transporters. Under these circumstances, we consult TCDB [38] as references and build a local version of transport information specific to modeling organisms. In our transport list, we also manually incorporate transport information from published genome-scale metabolic models and BIGG.

However, SEED and all published genome-scale metabolic models including those in BIGG only include information encoded in annotated genomes. None of these data resources includes spontaneous reactions, which happen without help from any gene product, therefore we accumulate a set of 336 spontaneous reactions listed with KEGG reaction identifiers, and insert them into models automatically.

**Table 3. Example of Transport Reactions Created in SWARM**

| BIGG | | SWARM | |
|---|---|---|---|
| **ENTRY** | **EQUATION** | **ENTRY** | **EQUATION** |
| T00021 | E02917 <=> C02917 | 12PPDt | 12ppd-S[e] <=> 12ppd-S[c] |

2) Issues of collecting data from frequently updated sources:

Databases we extract data from, especially SEED and KEGG, are actively expanding and updating. Therefore, we need to pump in this updated information into SWARM, and reflect these updates in the computational and modeling process of the workflow. As SEED provides APIs to access services provided by SEED and SWARM is built to integrate with SEED, we develop a collection of tools that use SEED APIs to extract data. Periodically, we re-run these tools to incorporate updates from SEED. In this scenario, SWARM can be viewed as a downstream of SEED and together with SEED, SWARM completes the Chicago Systems Biology Workflow as illustrated in Figure 1.

KEGG provides a set of WSDL APIs by building an API server using the SOAP technology [23]. These APIs enable users to write code that extracts data from KEGG automatically. Although WSDL is convenient to access remote data, it is rather slow to retrieve large amount of data. Using WSDL APIs KEGG provides, it takes days and days to retrieve the whole database with 2,912,739 genes, 15,050 compounds, 7,521 reactions, 71,826 pathways, 10,705 KO groups, and other information in 55 eukaryotes, 588 bacteria and 49 archaea. This large volume of data is updated frequently and the update affects data manipulations in SWARM. Therefore, In order to speed up the data retrieving process and reflecting the frequent update from KEGG, we use Rsync [50], a tool that provides fast incremental file transfer, to access their ftp server and download up-to-date data in KEGG. Afterwards, a series of tools are re-run to update mapping tables and other local data. More effective and probably more intelligent updating mechanisms are under investigation to immediately incorporate updates in SWARM.

Less frequently but from time to time, they are new genome-scale metabolic models published. We extract data from published results and BIGG, and parse out the information needed in SWARM.

Version control systems are introduced in the SWARM workflow to store and retrieve all versions of data in the repository. Together with data provenance mechanism to be discussed in Item 3), it is possible to identify and propagate changes throughout SWARM.

3) Data Provenance:

Data provenance is the derivation history of a data product, starting from its origin sources [49]. Achieving data provenance is an essential task in a dynamic scientific workflow. In SWARM, versions of code, parameters, resource versions, inputs and outputs to various tools, and auxiliary information are recorded by

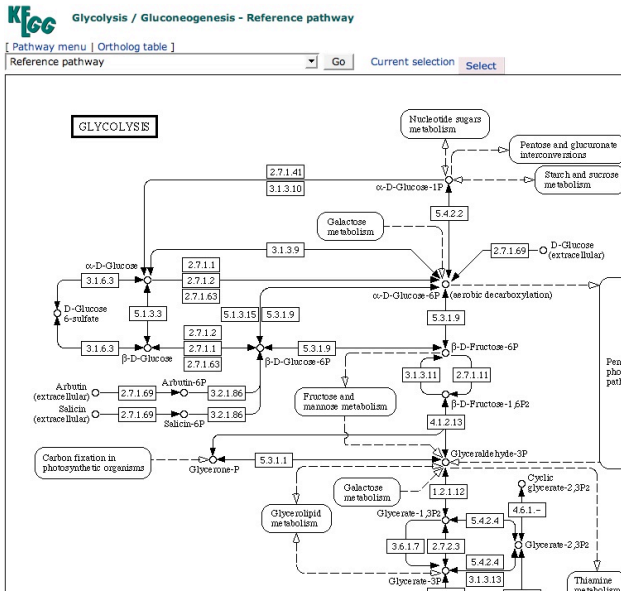


**Figure 3. Part of KEGG reference pathway map**

log files. The logging mechanism enables simple but effective data provenance.

4) Preprocess of data:

Besides building mapping tables as stated in Item 1) in this Subsection, it is desirable to preprocess some of the data considering large volume of data, frequent exceptions and computational complexity. Preprocessing of data occurs in the whole process of SWARM and we list three primary types here.

a) Balance reactions: Reaction lists in KEGG contain unbalanced reactions in mass and/or charge. In order to generate correct stoichiometry for modeling, mass and charge balance of these reactions is preferred. Hence, we develop a set of tools to balance KEGG reactions. The main procedure is as follows: For each reaction, its reaction formula is parsed to generate a list of reactants and products; The total charge/mass of reactants and the total charge/mass of products are compared and the difference is calculated; Then search through the compound list in KEGG and a matching compound is inserted into the reaction formula. In most cases, the charge unbalance of a reaction is caused by the missing proton (H+), while the missing of water molecule (H2O) leads to the mass unbalance of a reaction. There are cases where no matching compound can be inserted into the reaction formula to make it balanced. Manual work is carried out in these cases.

b) Break networks into pathway segments: A key strength of SWARM is the ability of filling network holes to extend metabolic models. In order to find candidate reactions to fill network holes, a set of predictors based on various evidences are built. These evidences form a hierarchy of gene-level, network-level to topology-level evidences. At the network-level, we have four different types of networks that are under investigation. The first type of data is KEGG reference pathway map, part of which is illustrated in Figure 3. This reference pathway map captures all known possible biochemical reactions in KEGG. The second kind of data is the organism's specific pathway map that is composed of reactions mapped by KO in KEGG, and there are totally 758 such maps. The third group of data is the draft reconstruction

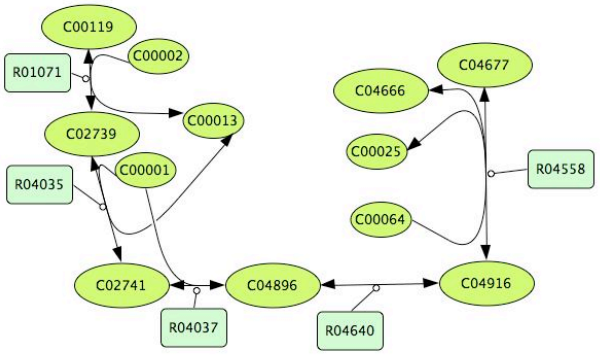


**Figure 4. A simple example part of a metabolic network**

from scenarios [26] for each complete genome in SEED, and there are 558 such maps. The fourth type of data includes two reconstructed genome-scale metabolic models available from published data.

Each of these networks is broken into pathway segments with 6 or less than 6. For example, a small network as shown in Figure 4 can be broken into a series of pathway segments with length of 2 to 6. Table 4 lists part of these pathway segments leading with starting compound identifier.

Then the number of reaction pairs in these pathway segments for every network-level evidence is calculated. As shown in Table 5, for KEGG reference pathway map with 4953 reactions, there are 148,377 pathway segments with length from 2 to 6. Therefore instead of reading the map, breaking it into segments and handling these segments at each run of the workflow, we preprocess pathway maps, parse the segments and save the intermediate data at retrievable places. The same type of preprocessing is performed for KEGG modules, draft metabolic models and published models.

c) Compute gene co-occurrence profiles: Gene co-occurrence profiles, which indicates the co-occurrence of gene pairs in SEED are used to build predictors at gene-level. We extract all the gene/protein families (noted as FigFams) in SEED and calculate co-occurrences of FigFam pairs across all complete bacteria genomes. From SEED, we extract 98, 850 Figfams and 449 complete bacteria genomes. Therefore, we have to compute computational probabilities of gene pairs by (98, 850 × 98, 850) that co-occur in 449 organisms in SEED. As shown in Table 6, we have a matrix with the size of 98, 850 by 449. Each genome has an entry in this matrix, and the value is 1 if a FigFam occurs in this genome and 0 otherwise. To calculate the probability of (98, 850 × 98, 850) gene pairs, we have to build a nested loop with three levels, each containing respectively 98850, 98850 and 449

**Table 4. Part of Pathway Segments from the Network in Figure 4.**

| Leading Compounds | Pathway Segments |
|---|---|
| C00119 | R01071 R04035; |
| C00119 | R01071 R04035 R04037; |
| C00119 | R01071 R04035 R04037 R04640; |
| C00119 | R01071 R04035 R04037 R04640 R04558; |
| C02739 | R04035 R04037; |
| … | … |

**Table 5. General Statistics of the KEGG**

| # genes | # KOs | # reactions | # compounds | # reactions in reference map | # reactions in modules | # reaction pairs in segments of reference map | # reaction pair s in segments of modules |
|---|---|---|---|---|---|---|---|
| 2,912,739 | 10,705 | 7,521 | 15,050 | 4953 | 1353 | 148377 | 36271 |

steps. Even with the help of sparse matrix manipulation, this calculation process still takes hours to complete and generates approximately 40 gigabytes of data. So calculating gene co-occurrence on the fly is significantly slow and we precompute this information for later use. Currently, this computation is achieved on a single machine but it is under investigation to perform this computation and other computation-intensive tasks on distributed systems such as TeraGrid and supercomputers such as BlueGene.

By preprocessing data, we not only speed up later process and remove repeated data handling, but also detect and deal with exceptions as early as possible.

5) Exception Handling:

Whenever exceptions happen, we filter out and record them. With the help of assistant tools, manual work is applied to check and solve these exceptions. In this process, domain knowledge of biology and/or chemistry is needed. For example, there are 232 BIGG reactions in E. coli model [5] that are not mapped to KEGG reactions using parsing tools as discussed in Item 1). To map this set of reactions as best as possible, we design tools that parse out reaction formulas of BIGG reactions, generate reactant compound sets and product compound sets for reaction formulas from two formats, reconstruct BIGG reaction formulas with BIGG compound abbreviations replaced by corresponding KEGG compound identifiers, compare with KEGG reaction formulas to search for the most similar form of KEGG reaction reactions. If an exact form of some KEGG reaction exists for this BIGG reaction, we add this KEGG-BIGG reaction association to our mapping table. If no exact match exists, we look at the reaction and investigate at corresponding functional role, EC number, KEGG pathway map to find an appropriate mapping of this BIGG reaction to a KEGG reaction with confidence of different level, and record this confidence in reaction mapping table. We then filter out those reactions with high confidences to be used in SWARM. The process above generates 122 mappings and after that, it leaves us 120 BIGG reactions that have no matched KEGG reactions. Currently, we leave these 120 BIGG reactions out of the reconstructed E. coli model and further investigation is needed to match these reactions.

# 5. VALIDATION

Testing and validation are essential steps to build models with the help of a scientific workflow. Therefore, it is necessary and valuable to validate SWARM by testing. Our validation mechanism includes experimenting with testing data including examples of different sizes. These examples involve approximately 1000 synthetic examples with 3~10 reactions and two reconstructions of published models [5,7] with hundreds of reactions. Results gained from these experiments will help refine SWARM. After training on these examples and known models, we extend draft models for those organisms without published models, with the goal of testing prediction mechanisms and SWARM.

**Table 6. Part of Gene Co-occurrence Matrix in SEED**

| | Figfam1 | Figfam2 | … | Figfam 98, 850 |
|---|---|---|---|---|
| Genome 1 | 1 | 0 | … | 0 |
| Genome 2 | 1 | 0 | … | 1 |
| Genome 3 | 1 | 1 | … | 0 |
| … | … | … | … | … |
| Genome 449 | 1 | 1 | … | 1 |

# 6. DISCUSSIONS AND SUMMARY

It is a challenge to automatically improve genome-scale metabolic models, due to the complexity of these models and the large volume of related information embedded in various data resources. However, there is an increasing need to extend genome-scale metabolic models for each organism with annotated genome, with the exponential growth of complete genomes. Our efforts to handle this problem lead to the design of SWARM, a scientific workflow for supporting the extension of genome-scale metabolic models. SWARM takes input from various databases, and generates metabolic models that can be simulated in different mathematical simulation environments.

Our contributions include building a scientific workflow that allows automatic construction of genome-scale metabolic models, a set of tools including a set of predictors based on various evidences and an ensemble of reaction predictors that can be used to improve metabolic models. These genome-scale metabolic models can be used to assemble components in the genome sequences, study how organisms behave under different situations, and thus perform systemic analysis of organisms to shed light on genotype-phenotype relationships.

After the development of SWARM is completed and mature, we plan to generalize all of the components and implant the entire workflow to the Swift workflow system. It is then possible to perform computation-demanding gene knockout experiments on a large number of metabolic models by running on large-scale distributed systems like TeraGrid and supercomputers such as BlueGene. Future work also includes the update of data from various databases more automatically, report exceptions and reflect the update in following executions in SWARM.

From our experiences working with the SWARM workflow, we find that there is a collection of issues that should attract more attention and get addressed better. For example, due to frequent update of biological and bioinformatics databases, it is demanding that version control, exception handling, data and information provenance be achieved more elegantly. These issues are extremely important in building bioinformatics workflows. Therefore, computer science, bioinformatics, and systems biology communities need to work together to address this data complexity problem. We believe that an automatic workflow from database to model is an important step in dealing with complex data. The availability of a large number of metabolic models will

lead to a new generation of important biological hypotheses and experimental designs based on the analysis of these models.

## 7. ACKNOWLEDGEMENTS

The authors would like to thank Federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Contract No. HHSN266200400042C. This work was supported in part by the U.S. Dept. of Energy under Contract DE-AC02-06CH11357.The authors are grateful for discussions and/or share of data from Ross Overbeek, Terry Disz, Matthew Cohoon, Matthew DeJongh, Christopher Henry, Michael Kubal, Wenjun Wu, Fangfang Xia, and Jenifer Zinner. The authors thank SEED annotators, and the SEED development group for providing access to SEED and support in development within this environment. The authors greatly appreciate reviewers for their time and invaluable suggestions.

## 8. REFERENCES

[1] Reed, J.L. and Palsson, B.Ø. 2003. Thirteen Years of Building Constraint-Based In Silico Models of Escherichia coli. Journal of Bacteriology, Vol. 185, No. 9, p. 2692-2699.

[2] Edwards, J.S., Covert, M., and Palsson, B.Ø. 2002. Metabolic modelling of microbes: the flux-balance approach. Environ. Microbiol. 4:133-140.

[3] Varma, A. and Palsson, B.Ø. 1994. Metabolic flux balancing: basic concepts, scientific and practical use. Bio/Technology 12:994-998.

[4] Feist, A.M., Henry, C.S., *et al.* 2007. A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Molecular Systems Biology.

[5] Reed, J.L., Vo, T.D., Schilling, C.H., and Palsson, B.Ø. 2003. An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR). Genome Biol. 4(9):R54.

[6] Edwards, J.S. and Palsson, B.Ø. 2000. The Escherichia coli MG1655 in silico metabolic genotype: its definition, characteristics, and capabilities. Proc Natl Acad Sci USA. 97:5528–5533.

[7] Becker, S.A. and Palsson, B.Ø. 2005. Genome-scale reconstruction of the metabolic network in Staphylococcus aureus N315: an initial draft to the two-dimensional annotation. BMC Microbiol. 7;5(1):8.

[8] Thiele, I., Vo, T.D., Price, N.D., and Palsson, B.Ø. 2005. Expanded Metabolic Reconstruction of Helicobacter pylori (iIT341 GSM/GPR): an In Silico Genome-Scale Characterization of Single- and Double-Deletion Mutants. Journal of Bacteriology, Vol.187, No.16, p.5818–5830.

[9] Forster, J., Famili, I., Fu, P., Palsson, B.Ø., and Nielsen, J. 2003. Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network. Genome Res. 13(2):244-53.

[10] Duarte, N.C., Herrgard, M.J., and Palsson, B.Ø. 2004. Reconstruction and Validation of Saccharomyces cerevisiae iND750, a Fully Compartmentalized Genome-Scale Metabolic Model. Genome Research 14:1298-1309.

[11] Oliveira, A.P., Nielsen, J., and Forster, J. 2005. Modeling Lactococcus lactis using a genome-scale flux model. BMC Microbiol. 27;5:39.

[12] Oh, Y.K., Palsson, B.Ø., Park, S.M., Schilling, C.H., and Mahadevan, R. 2007. Genome-scale Reconstruction of Metabolic Network in Bacillus subtilis Based on High-throughput Phenotyping and Gene Essentiality Data. J Biol Chem. 10.1074.

[13] Schilling, C.H., Covert, M.W., Famili, I., Church, G.M., Edwards, J.S., and Palsson, B.Ø. 2002. Genome-scale metabolic model of Helicobacter pylori 26695. J Bacteriol. 184(16):4582-93.

[14] Gates, B., Pinchuk, G.E., Schilling, C., *et al.* 2006. Genome-Scale Metabolic Model of Shewanella oneidensis MR1. GTL.

[15] Feist, M.A., Scholten, C.J., Palsson, B.Ø., et.al. 2006. Modeling methanogenesis with a genome-scale metabolic reconstruction of Methanosarcina barkeri. Molecular Systems Biology.

[16] Edwards, J.S. and Palsson, B.Ø. 1999. Systems Properties of the Haemophilus influenzae Rd Metabolic Genotype. Journal of Biological Chemistry, 274, 17410-17416.

[17] Duarte, N.D., Becker, S.A., *et al.* 2007. Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proc Natl Acad. Sci USA 104(6):1777-82.

[18] BIGG (A Biochemical Genetic and Genomic Database of large scale metabolic reconstructions.): http://bigg.ucsd.edu/

[19] Osterman, A. and Overbeek, R. 2003. Missing genes in metabolic pathways: a comparative genomics approach. Curr Opin Chem Biol , 7:238-251.

[20] Overbeek, R., Disz, T., and Stevens, R. 2004. The SEED: a peer-to-peer environment for genome annotation. Communications of the ACM, Vol. 47, No. 11, Pages 46-51

[21] The SEED: an Annotation/Analysis Tool Provided by FIG: http://theseed.uchicago.edu/.

[22] Kanehisa, M., Araki, M., *et al.* 2008. KEGG for linking genomes to life and the environment. Nucleic Acids Res. 36, D480-D484.

[23] KEGG: Kyoto Encyclopedia of Genes and Genome: http://www.genome.jp/kegg/.

[24] Kharchenko, P., Vitkup, D., and Church, G.M. 2004. Filling gaps in a metabolic network using expression information. Bioinformatics, 20(Suppl 1):I178-I185.

[25] Kharchenko, P., Chen, L., *et al.* 2006. Identifying metabolic enzymes with multiple types of association evidence. BMC Bioinformatics. 29;7(1):177.

[26] DeJongh, M., Formsma, K., Boillot, P., Gould, J., Rycenga, M., and Best, A. 2007. Toward the automated generation of genome-scale metabolic networks in the SEED. BMC Bioinformatics, 8:139.

[27] Becker, S.A., Feist, A.M., *et al.* 2007. Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox. Nature Protocols 2, - 727 - 738 .

[28] SimPheny: www.genomatica.com/solutions_simpheny.shtml.

[29] Klamt, S., Stelling, J., Ginkel, M., and Gilles, E.D. 2003. FluxAnalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. Bioinformatics, 19(2): 261-269.

[30] Klamt, S., Saez-Rodriguez, J., and Gilles, E.D. 2007. Structural and functional analysis of cellular networks with CellNetAnalyzer. BMC Systems Biology, 1:2.

[31] Green, M.L. and Karp, P.D. 2004. A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. BMC Bioinformatics, vol. 5, no. 76.

[32] Karp, P.D., Paley, S., and Romero, P. 2002. The Pathway Tools software. Bioinformatics. 18 Suppl 1:S225-32.

[33] Overbeek, R., Begley, T., Butler, R.M., *et al.* 2005. The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. Nucleic Acids Res. 7;33(17):5691-702.

[34] Graphviz: Graph Visualization Software: www.graphviz.org.

[35] CellDesigner: A modeling tool of biochemical networks: http://www.celldesigner.org/

[36] Systems Biology Markup Language (SBML): www.sbml.org.

[37] Gene Ontology: http://www.geneontology.org/

[38] TCDB: Transport Classification Database: www.tcdb.org.

[39] Foster, I. and Kesselman, C. (editors), 1999. The Grid: Blueprint for a Future Computing Infrastructure. Morgan Kaufmann Publishers, USA.

[40] Yu, J. and Buyya, R. 2005. A Taxonomy of Scientific Workflow Systems for Grid Computing. SIGMOD Record, Vol. 34, No. 3.

[41] Barker, A. and Hemert, J. 2007. Scientific Workflow: A Survey and Research Directions. In Proceedings of the The Third Grid Applications and Middleware Workshop (GAMW'2007), Gdansk, Poland.

[42] Ludäscher, B., Altintas, I., *et al.* 2005. Scientific Workflow Management and the Kepler System. Concurrency and Computation: Practice & Experience, 36.

[43] Bowers, S. and Ludascher, B. 2005. Actor-Oriented Design of Scientific Workflows. In 24 th Intl. Conf. on Conceptual Modeling (ER).

[44] Oinn, T., Greenwood, M., *et al.* 2005. Taverna: lessons in creating a workflow environment for the life sciences. Concurrency and Computation: Practice and Experience, Vol 18, Issue 10, Pages 1067 – 1100.

[45] Stevens, R.D., Robinson, A.J., and Goble, C.A. 2003. myGrid: personalised bioinformatics on the information grid. Bioinformatics 19(1) c Oxford University Press.

[46] Rygg, A., Roe, P., Wong, O., and Sumitomo, J. 2008. GPFlow: An Intuitive Environment for Web Based Scientific Workflow. Concurrency and Computation: Practice and Experience, Vol 20, Issue 4, pp. 393 - 408.

[47] Merelli, I., Morra, G., and Milanesi, L. 2005. Bioinformatics Workflow using ASSIST on GRID. In Proc. of The Network Tools and Applications in Biology Workshop (NETTAB), Naples, Italy.

[48] Swift: http://www.ci.uchicago.edu/swift/.

[49] Simmhan, Y., Plale, B., and Gannon, D. 2005. A Survey of Data Provenance in e-Science, SIGMOD Record, Vol. 34, No. 3.

[50] Rsync: http://samba.anu.edu.au/rsync/.