2016

Computational methods for integrated analysis of omics and pathway data

Jesse R. Walsh
Iowa State University

Recommended Citation

Walsh, Jesse R., "Computational methods for integrated analysis of omics and pathway data" (2016). *Graduate Theses and Dissertations*. 15198.
https://lib.dr.iastate.edu/etd/15198

Computational methods for integrated analysis of omics and pathway data

by

Jesse R. Walsh

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Bioinformatics and Computational Biology

Program of Study Committee:

Julie Dickerson, Co-major Professor

Carolyn Lawrence-Dill, Co-major Professor

Laura Jarboe

Philip Dixon

Taner Sen

Iowa State University

Ames, Iowa

2016

DEDICATION

I would like to dedicate this work to my wife Sheila and to my son Nathaniel. They have shared every triumph and sacrifice with me throughout this journey, and it has only been with their love and perseverance that this work has been made possible. I consider my family to be both my greatest achievement and my greatest treasure. I would also like to dedicate this work to my parents, Allyn and Sheila Walsh. They have provided me with a lifetime of loving guidance, without which I wouldn't be the person I am today. I will always be as proud of them as they are of me. Finally, I would like to thank my many friends and family for their love and support over the years. In particular, to the memory of my grandparents Betty and Roger Walsh, with whom I had grown especially close in recent years. I know their love will be with us always.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who have helped me in various ways throughout the course of my research. First and foremost, Dr. Julie A. Dickerson for her guidance, patience and support throughout this research and the writing of this thesis. Her insights and expertise have been invaluable to the advancement of my work. I would like to thank my co-advisor Dr. Carolyn Lawrence, who often provided excellent feedback and helped to set me on the path to success. I would also like to thank my committee members for their efforts and contributions: Dr. Taner Sen., Dr. Laura Jarboe, and Dr. Phil Dixon. In particular, Dr. Taner Sen has been an invaluable collaborator and his advice and guidance have been instrumental to my success.

I would like to thank the members of the Dickerson lab both past and present, many of whom have contributed to my success over the years. Dr. John Van Hemert helped me in my early years to learn many of the necessary skills to become a successful researcher. I would like to thank the many members of CBiRC and MaizeGDB who have provided advice and feedback on many aspects of my work. Finally, I would like to thank Trish Stauble for her advice and coordination during my time as a student in the BCB program.

This work was funded primarily through a fellowship made possible by the National Science Foundation grant EEC-0813570. Additional funding was made available through a fellowship award under the Integrative Graduate Education and Research Training (IGERT) grants from the National Science Foundation.

ABSTRACT

One of the key tenets of bioinformatics is to find ways to enable the interoperability of heterogeneous data sources and improve the integration of various biological data. High-throughput experimental methods continue to improve and become more easily accessible. This allows researchers to measure not just their specific gene or protein of interest, but the entirety of the biological machinery inside the cell. These measurements are referred to as "omics", such as genomics, transcriptomics, proteomics, metabolomics, and fluxomics.

Omics data is highly interrelated at the systems-level, as each type of molecule (DNA, RNA, protein, etc.) can interact with and have an impact on the other types. These interactions may be direct, such as the central dogma of biology that information flows from DNA to RNA to protein. They may also be indirect, such as the regulation of gene expression or metabolic feedback loops. Regardless, it is becoming apparent that multiple levels of omics data must be analyzed and understood simultaneously if we are to advance our understanding of systems-level biology.

Much of our current biological knowledge is stored in public databases, most of which specialize in a particular type of omics or a specific organism. Despite efforts to improve consistency between databases, there are many challenges which can impede efforts to meaningfully compare or combine these resources. At a basic level, differences in naming and internal database ID assignments prevent simple mapping between objects in these databases. More fundamentally, though, is the lack of a standardized way to define equivalency between two functionally identical biological entities.

One benefit of improving database interoperability is that targeted high quality data from one database can be used to improve another database. Comparison between MaizeCyc and CornCyc identified many manually curated GO annotations present in MaizeCyc but not in CornCyc. CycTools facilitates the transfer of high-quality annotation data from one database to another by automatically mapping equivalent objects in both databases. This java-based tool has a graphical user interface which guides users through the transfer process.

A case study which uses two independent *Zea Mays* pathway databases, CornCyc and MaizeCyc, illustrates the challenges of comparing the content of even closely related resources. This example highlights the downstream implications that the choice of initial computational enzymatic function assignment pipelines and subsequent manual curation had on the overall scope and quality of the content of each database. We compare the prediction accuracy of the protein EC assignments for 177 maize enzymes between these resources and find that while MaizeCyc covers a broader scope of enzyme predictions, CornCyc predictions are more accurate.

The advantage of high quality, integrated data resources must be realized through analysis methods which can account for multiple data types simultaneously. Due to the difficulty in obtaining systems-wide metabolic flux measurements, researchers have made several efforts to integrate transcriptional regulatory data with metabolic models in order to improve the accuracy of metabolic flux predictions. Transcriptional regulation involves the binding of transcription factors (i.e. proteins) to binding sites on the DNA in order to positively or negatively influence expression of the targeted gene. This has an indirect, downstream impact on the organism's metabolism, as metabolic reactions depend on gene-derived enzymes in order to catalyze the reaction.

A novel method is proposed which seeks to integrate transcriptional regulation and metabolic reactions data into a single model in order to investigate the interactions between metabolism and regulation. In contrast to existing methods which seek to use transcriptional regulation networks to limit the solution space of the constraint-based metabolic model, we seek to define a transcriptional regulatory space which can be associated with the metabolic distribution of interest. This allows us to make inferences about how changes in the regulatory network could lead to improved metabolic flux.

1. GENERAL INTRODUCTION

Introduction

Continual improvements to high-throughput experimental technologies have led to a vast increase in publicly available biological data, fundamentally shifting how we approach biological problems. The "omics" data produced by these technologies encompass system-wide measurements of cellular biology, including genomics, transcriptomics, proteomics, metabolomics, and fluxomics. The availability of this systems-wide data allows researchers to study interactions between various components of an organism, rather than studying the individual components themselves.

Omics data can often be represented as a network of biological interactions. Biological networks might represent regulatory interactions, protein interactions, or metabolic pathways. Many public databases have been created to facilitate storage, browsing, and retrieval of this information. Some specialize in certain types of omics [1, 2], certain types of interactions [3], or specific organisms [4]. These databases allow researchers to access biological network data for use in many applications, such as protein and gene annotation, elucidating regulatory interactions, and metabolic engineering applications.

While the availability of these resources has been beneficial to the research community, there remain many challenges to using this vast wealth of knowledge effectively. The astute researcher may notice many inconsistencies between various public databases. Unfortunately, there has been surprisingly little work done to quantify or explain how these differences may affect results which are derived from such data. Part of the reason for this lies in the challenges inherent in comparing various resources. These challenges have been well documented [5, 6], but not resolved. Pathways are often defined using arbitrary guidelines for where to draw the pathway boundaries. Metabolites and chemical compounds can change structure (e.g. stereochemistry) and chemical formula under different conditions (e.g. pH), and are often given many synonyms. Typographical differences in names can add additional complexity, as some versions may spell out "alpha" while others may use the symbol "α" instead. Genes and proteins often owe their annotations to blast searches and gene homology, an effective but inexact method of comparing gene sequences.

Additionally, the provenance of the data in public databases can be unclear. Not all data is considered equal in value, and the impact of these different levels of data on the generation of our knowledge bases is poorly characterized. We consider three classes of data representing the methods which were used to generated it (Figure 1.1). Computational data is generated using methods such as homology transfer and machine learning and often includes a built-in acceptable error rate. Data generated from these methods are important, especially since they are often used to help fill in missing information for lesser studied organisms. However, as this data is typically inexpensive to reproduce, it is considered lower in value than other types. Experimental data represent measurements of biological processes such as transcriptomics and proteomics data, and as such is considered more valuable than computational predictions. Nevertheless, it is important to consider that the methods used to measure biological data can also be subject to significant quality issues as well. Curated data is generally considered the most valuable data, as it represents knowledge reviewed by a domain expert and backed by current literature. Unfortunately, this makes curated data both expensive and time-consuming to produce.

One of the challenges in dealing with biological networks involves representing specific differences in substrains of a particular organism and in handling context-specific data. In metabolic engineering applications, the goal of a study is to alter the systems level regulatory and metabolic functions of an organism in order to increase production or recovery of a desired metabolite. For example, changes may include the addition or removal of metabolic and/or regulatory functions. In Chapter 2, we describe a software tool which facilitates the modification of metabolic databases to match specific alterations in the organism represented. This tool automates many of the challenges in matching database objects and provides methods for conflict detection and resolution when multiple versions of the same information are present.

Chapter 3 describes two databases representing the same organism, *Zea Mays*. The databases, CornCyc [7] and MaizeCyc [8], are both derived from the same gene model set but use different computational pipelines to perform their enzymatic function annotations. This difference in annotation method led both databases to significantly different conclusions. These resources differ in gene and protein coverage, reaction and pathway content, and in distribution of reactions across EC categories. This study is used to illuminate some of the challenges in comparing public data. Despite using the same gene model set, each resource had different internal identifiers for their gene and protein names. Alternative splicing is represented through a naming convention rather than encoded in a standard way within each database. Pathways differences included both differences in pathway variants (pathways with similar functionality) as well as outdated versions of existing pathways. In addition to describing the general comparison and overlap of these resources, we validate 177 enzyme function annotations in order to illustrate the difference in accuracy between these resources.

Metabolic networks have been used in metabolic engineering to provide a framework for a system-wide integration of biochemical function and control information. Metabolic pathway networks, which are used to construct constraint-based metabolic models of biochemical networks, have received a great deal of attention in recent years due to the increasing availability of curated constraint-based metabolic models [9] and the difficulty in obtaining reliable and complete parameter sets necessary to construct kinetic models [10]. The primary advantage of constraint-based models is that they can be constructed using only reaction stoichiometry information for an organism. Several tools are available to aid in constraint-based network analysis, including COBRA ToolBox [11], CycSim [12], and CellNetAnalyzer [13].

In order to improve our understanding of the biological principles which govern the control and function of cellular metabolism, it is necessary to analyze not just the interactions within a single network, but the interactions between multiple levels of biological networks. In Chapter 4, we describe a novel method to combine the regulatory and metabolic networks of *E. coli* into a unified constraint-based model in order to investigate the constraints metabolic requirements place on cellular regulatory states.

Dissertation Organization

This manuscript is organized into five chapters. Chapters 2-4 are manuscripts that have either been published in a peer reviewed journal or are being submitted to one. Chapter 5 is a general discussion on the significance and impact of the studies presented in Chapters 2-4.

Chapter 2, A Computational Platform to Maintain and Migrate Manual Functional Annotations for BioCyc Databases (published in BMC Systems Biology, Chapter 2) describes a software tool which can assist users to map biological annotations between BioCyc databases. This chapter is significant as it addresses a practical need to be able to improve existing resources based either on high-quality data found at an external source, or to provide context-specific annotations for use by researchers interested in novel variations of an otherwise well-characterized organism. We demonstrate the utility of this software by automating the copy and transfer of high-quality GO annotations from one database to another. My contribution to this work was to identify and map the high-quality GO annotations, to design and develop the software, and to perform the transfer of annotation data. I prepared all figures and wrote the initial draft of the manuscript.

Chapter 3, The Quality of Metabolic Pathway Resources Depends on Initial Enzymatic Function Assignments and Level of Manual Curation: A Case for Maize (to be submitted to Plant Physiology, Chapter 3) presents a comparison between two resources for the organism *Zea Mays* (corn). Two separate groups have created unique databases for the same organism using the same starting gene model. By utilizing different protein prediction algorithms and applying varying levels of curation, the resulting databases are significantly different. This chapter is significant in that it illuminates some of the challenges for interoperability between heterogeneous biological resources, as well as characterizes the impact differences in content can have on analysis. My part in this work was to provide the database structure and access expertise, design and develop the software that performed the comparison, and generate the

EC distribution component. I prepared all figures and wrote the initial draft of the manuscript with input from my co-authors.

Chapter 4, Modeling the Effect of Metabolic Constraints on Transcription Factor Activity Levels (a manuscript prepared for submission to a scholarly journal, Chapter 4), describes a method by which metabolic models of metabolism can be used to infer transcriptional regulator activities in *E. coli.* This work is significant as it describes a novel approach to computational strain design through the use of integrated regulatory and metabolic network models. In contrast to existing methods which seek to use transcriptional regulation networks to limit the solution space of the constraint-based metabolic model, we seek to define a transcriptional regulatory space which can be associated with the metabolic distribution of interest. My contribution to this work was to initially conceive and design the method for integrating the two data types, perform the data processing and initial setup to generate the models using existing software, and to test and validate the method. I prepared all figures and wrote the initial draft of the manuscript.

Finally, Chapter 5 discusses general conclusions which can be drawn from chapters 2-4. I discuss the significance and impact of each work, and identify key future directions which can build upon the work done here.

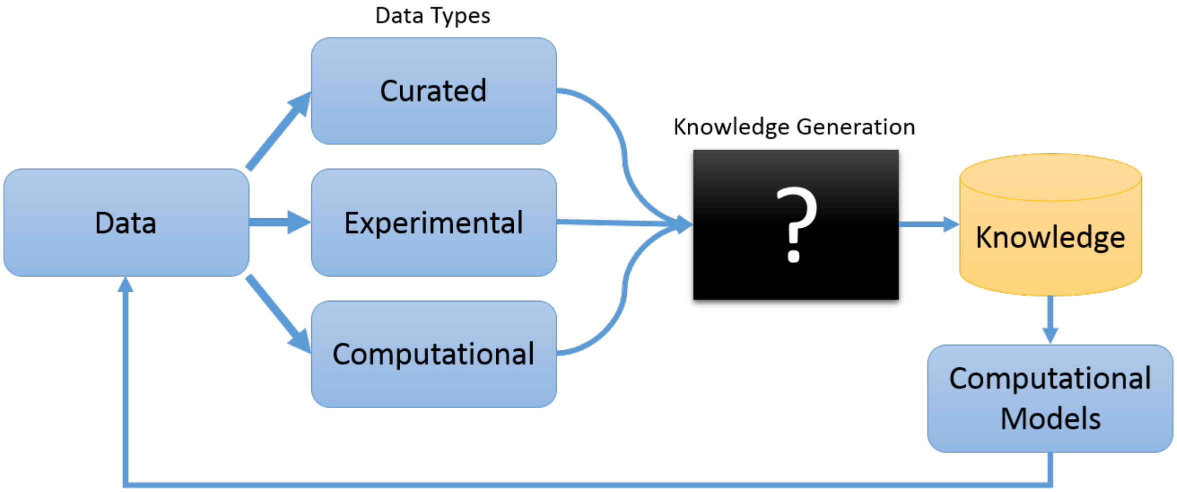


Figure 1.1 Biological data is understood to be inherently different in quality depending on its source. Computationally derived knowledge is relatively easy to generate but is understood to have a built-in level of acceptable error. Experimental data represent actual biological measurements, and as such are given greater consideration. However, methods which characterize biological processes are also prone to measurement errors. Curated data is both time-consuming and expensive to generate as it requires a domain expert to review all available literature to ensure a high level of accuracy in the data. These data are used to generate our knowledge bases. Unfortunately, the methods which combine this knowledge are often poorly characterized and under documented. Models derived from knowledge bases are used to generate more data which can serve as a method for the iterative refinement of our biological knowledge.

Bibliography

[1] Consortium, T.U.: UniProt: a hub for protein information. Nucleic Acids Research **43**(D1), 204–212 (2015). doi:10.1093/nar/gku989

[2] Faith, J.J., Driscoll, M.E., Fusaro, V.A., Cosgrove, E.J., Hayete, B., Juhn, F.S., Schneider, S.J., Gardner, T.S.: Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata. Nucleic Acids Research **36**(Database issue), 866–870 (2008). doi:10.1093/nar/gkm815

[3] Salgado, H., Peralta-Gil, M., Gama-Castro, S., Santos-Zavaleta, A., Muiz-Rascado, L., Garca-Sotelo, J.S., Weiss, V., Solano-Lira, H., Martnez-Flores, I., Medina-Rivera, A., Salgado-Osorio, G., Alquicira-Hernndez, S., Alquicira-Hernndez, K., Lpez-Fuentes, A., Porrn-Sotelo, L., Huerta, A.M., Bonavides-Martnez, C., Balderas-Martnez, Y.I., Pannier, L., Olvera, M., Labastida, A., Jimnez-Jacinto, V., Vega-Alvarado, L., Del Moral-Chvez, V., Hernndez-Alvarez, A., Morett, E., Collado-Vides, J.: RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. Nucleic acids research **41**(Database issue), 203–213 (2013). PMID: 23203884 PMCID: PMC3531196

[4] Keseler, I.M., Collado-Vides, J., Santos-Zavaleta, A., Peralta-Gil, M., Gama-Castro, S., Muniz-Rascado, L., Bonavides-Martinez, C., Paley, S., Krummenacker, M., Altman, T., Kaipa, P., Spaulding, A., Pacheco, J., Latendresse, M., Fulcher, C., Sarker, M., Shearer, A.G., Mackie, A., Paulsen, I., Gunsalus, R.P., Karp, P.D.: EcoCyc: a comprehensive database of Escherichia coli biology. Nucleic Acids Research **39**(Database), 583–590 (2010). doi:10.1093/nar/gkq1143

[5] Wittig, U., Beuckelaer, A.D.: Analysis and comparison of metabolic pathway databases. Briefings in Bioinformatics **2**(2), 126–142 (2001). doi:10.1093/bib/2.2.126

[6] Altman, T., Travers, M., Kothari, A., Caspi, R., Karp, P.D.: A systematic comparison of the MetaCyc and KEGG pathway databases. BMC Bioinformatics **14**(1), 112 (2013). doi:10.1186/1471-2105-14-112

[7] Chae, L., Lee, I., Shin, J., Rhee, S.Y.: Towards understanding how molecular networks evolve in plants. Current Opinion in Plant Biology **15**(2), 177–184 (2012)

[8] Monaco, M.K., Sen, T.Z., Dharmawardhana, P.D., Ren, L., Schaeffer, M., Naithani, S., Amarasinghe, V., Thomason, J., Harper, L., Gardiner, J., Cannon, E.K., Lawrence, C.J., Ware, D., Jaiswal, P.: Maize metabolic network construction and transcriptome analysis. The Plant Genome **6**(1), 0 (2013)

[9] Medema, M.H., van Raaphorst, R., Takano, E., Breitling, R.: Computational tools for the synthetic design of biochemical pathways. Nat Rev Micro **10**(3), 191–202 (2012). doi:10.1038/nrmicro2717

[10] Orth, J.D., Thiele, I., Palsson, B..: What is flux balance analysis? Nature Biotechnology **28**(3), 245–248 (2010). doi:10.1038/nbt.1614

[11] Schellenberger, J., Que, R., Fleming, R.M.T., Thiele, I., Orth, J.D., Feist, A.M., Zielinski, D.C., Bordbar, A., Lewis, N.E., Rahmanian, S., Kang, J., Hyduke, D.R., Palsson, B..: Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nature Protocols **6**(9), 1290–1307 (2011). doi:10.1038/nprot.2011.308

[12] Le Fvre, F., Smidtas, S., Combe, C., Durot, M., d'Alch-Buc, F., Schachter, V.: CycSiman Online Tool for Exploring and Experimenting with Genome-Scale Metabolic Models. Bioinformatics **25**(15), 1987–1988 (2009). doi:10.1093/bioinformatics/btp268

[13] Klamt, S., Saez-Rodriguez, J., Gilles, E.D.: Structural and functional analysis of cellular networks with CellNetAnalyzer. BMC Systems Biology **1**(1), 2 (2007). doi:10.1186/1752-0509-1-2

2. A COMPUTATIONAL PLATFORM TO MAINTAIN AND MIGRATE MANUAL FUNCTIONAL ANNOTATIONS FOR BIOCYC DATABASES

A paper published in BMC Systems Biology

Jesse R. Walsh, Taner Z. Sen, and Julie A. Dickerson

Abstract

Background

BioCyc databases are an important resource for information on biological pathways and genomic data. Such databases represent the accumulation of biological data, some of which has been manually curated from literature. An essential feature of these databases is the continuing data integration as new knowledge is discovered. As functional annotations are improved, scalable methods are needed for curators to manage annotations without detailed knowledge of the specific design of the BioCyc database.

Results

We have developed CycTools, a software tool which allows curators to maintain functional annotations in a model organism database. This tool builds on existing software to improve and simplify annotation data imports of user provided data into BioCyc databases. Additionally, CycTools automatically resolves synonyms and alternate identifiers contained within the database into the appropriate internal identifiers.

Conclusions

Automating steps in the manual data entry process can improve curation efforts for major biological databases. The functionality of CycTools is demonstrated by transferring GO term annotations from MaizeCyc to matching proteins in CornCyc, both maize metabolic pathway databases available at MaizeGDB, and by creating strain specific databases for metabolic engineering.

Background

Lower costs in genomic sequencing and improved methods of generating computationally predicted functional annotations has led to the development of many model organism databases using the BioCyc framework [1]. While computationally derived draft model organism databases provide useful starting points for storing biological knowledge, computationally predicted annotations are known to suffer from significant false negative rates [2]. The accuracy of annotations can be substantially improved by providing manual annotations mined from literature by expert curators. Unfortunately, manual curation efforts have not kept up with the proliferation of new databases. There are currently over 3500 databases in the BioCyc collection, however only 42 of these currently receive moderate or intensive manual review [3].

Among the databases which receive manual review, maintaining manually curated data can present a challenge. When an improved reference sequence is released for an organism, the BioCyc database representing that organism must be recreated in order to incorporate the new sequence data. While computationally predicted annotations within the database should be updated using the new input data, it is usually preferred to keep existing manual annotations even if the computational annotations are more recent. There is a need for tools which can assist curators in persisting manually curated data through the update process either through automation or by providing pipelines for the transfer of manual annotations of these databases. Additionally, when several distinct databases host biological data for the same organism, it is desirable to share manually curated annotations between these databases in order to improve data accuracy without duplicating curator efforts. In order to facilitate the transfer of data between databases, robust import and export features must be made available.

Pathway Tools [4], the software which supports development and management of BioCyc databases, provides several options for updating a BioCyc database. Changes may be made manually within the pathway tools software by first locating the object to update and then entering edit mode to make the changes to that object, as shown in Figure 2.1. Each object type (protein, gene, metabolite, etc.) has a specific data entry form, which can be filled out and saved. While this method allows the curator to directly review and verify the changes entered into the database, it is inefficient when performing large numbers of updates.

Pathway Tools supports data imports through two file formats, "spreadsheet format" or "Lisp-format". Examples are provided in the supplemental materials C.1. The spreadsheet format imports are limited in that some data cannot be imported using this method, including GO term annotations, stoichiometry, and cellular localization. While the Lisp-format supports the import of these data types, it requires users to have an understanding of the Lisp data structure implemented in the BioCyc framework and is not easily converted to other standard formats.

A final import option provided by Pathway Tools is through an application programming interface (API) which exposes low level access to the BioCyc data structure. The API is very flexible in that users can design queries to suit their specific needs, but they must have a detailed understanding of the internal structure of a BioCyc database in order to do so. Certain modifications to a BioCyc database, such as GO term annotations, require additional steps in order to maintain the referential integrity of the database. This provides further barriers to use, as users must have an understanding of how Pathway Tools implements storage of these features.

Despite the diversity of import methods provided by Pathway Tools, there remains a need for an import pipeline which is both capable of importing slot-value annotation data in batch and accessible to researchers who are not experts in programming or BioCyc database structure.

CycTools is a graphical interface for the BioCyc family of databases which improves data management by providing methods which can import slot-value annotation data in batch.

Implementation

CycTools Dependencies

BioCyc is a family of databases built using the BioCyc Framework. Each member database of the BioCyc collection typically represents the pathway and genomic data of a specific organism. BioCyc databases are built on the Frame Representation System (FRS) known as Ocelot [5], which extends the Generic Frame Protocol (GFP). The native storage format for BioCyc data is an object oriented database representation based on frames. The hierarchical nature of data represented in a frame can be seen in Figure 2.2. A frame is a high level container that groups information regarding either biological entities (genes, proteins, transcripts, compounds, etc.) or biological relationships (reactions, pathways, regulation, etc.). Information about the object a frame represents can be stored in either slots or slot-value-annotations. Information stored in slots describes the frame (i.e. the name of the object, its physical properties, or annotations assigned to it), while information in slot-value-annotations provides context for the information in the slots (i.e. pubmed citations, author credits, or experimental evidence codes). The data stored in frames and slots in the database can be accessed programmatically through the Pathway Tools API.

The API exposes many of the internal functions of Pathway Tools and allows low level access to the internal data structure of any BioCyc database hosted by Pathway Tools. Advanced users can create third-party software which can read or write to BioCyc databases using customized queries. The API is designed to support the Lisp programming language, but the libraries PerlCyc [6] and JavaCycO [7] allow users to access the API through Perl and Java respectively.

JavaCycO is an object-oriented improvement to the JavaCyc library. JavaCycO contains the JavaCyc [6] class and is fully backwards compatible with it. In addition to extending and improving the functionality of JavaCyc, JavaCycO provides a client-server model for accessing the Pathway Tools API. By running the server “JavaCycServer” on the same machine as

Pathway Tools, JavaCycO provides remote access to the Pathway Tools API to JavaCycO clients. CycTools depends on the JavaCycO library to provide access to the Pathway Tools API in order to read and write to a BioCyc database.

Cloning a Database

Generally speaking, CycTools can modify any BioCyc database hosted by Pathway Tools. Two notable exceptions to this are the MetaCyc and EcoCyc databases, which are integrated into Pathway Tools and flagged as read-only. Since these databases can not be removed or modified, the only way to edit them is to edit a copy. Pathway Tools will also refuse to load two databases with the same name, which prevents the user from simply installing a second copy of a database without first renaming and modifying several of the files and folders within the copy. This restriction will also prevent the user from creating and hosting several versions of a database in the same Pathway Tools instance. In order to circumvent this restriction, a bash script which automatically clones a database and modifies the appropriate files was created.

Overview of Import Process

The CycTools import function provides a graphical pipeline for importing spreadsheet data into frame objects in the Pathway Genome Database (PGDB). The import utility takes as input a comma-separated data file, maps the data to frames in the PGDB, previews the resulting changes to the PGDB, and performs the update of the PGDB as shown in Figure 2.3.

CycTools must be able to connect to a server running Pathway Tools in API mode and JavaCycO. Once connected, the user will select one of the available import types: import slot data, import slot-value annotation data, import GO annotations, delete frames, or create transcriptional regulation frames. This determines the format of the import file and how the imported data are applied to database objects. Additional options are available which allow the user to specify how to handle existing data in a slot or annotation which will be modified during import, shown in Figure 2.4.

If the overwrite option is set, CycTools will first delete the existing data in a slot or annotation before writing the user provided data to that slot or annotation. If the ignore duplicates option is set, CycTools will check each new value against each existing values in a slot or annotation. If the new value exactly matches an existing value, it will not be added to the slot or annotation. This option will prevent the user from adding a duplicate value to a slot or annotation, but will not remove an existing duplication. Thus, if a protein were to be annotated with a single GO term twice, this option will prevent CycTools from adding a third identical annotation using that GO term, but would leave the existing annotations.

The author credits option allows the user to assign credit to an individual or organization for each frame updated during the import process. CycTools autofills a list of curators and organizations described in the currently selected database. For each frame updated during the import, the frame is modified to append the curator or organization to the "CREDITS" slot. This update is annotated as a revision to the frame and is timestamped to the current system time.

GO Term Annotations

GO term annotation imports are handled slightly different from other annotations imports. In particular, Pathway Tools has specific requirements for the storage of GO term descriptions within a BioCyc database. The Pathway Tools API provides a method called "import-go-terms" which automatically creates the necessary frames when provided with a valid GO term. Pathway Tools is packaged with a file containing GO term information which is used by this method to populate the GO term frames it creates. CycTools makes a call to "import-go-terms" once for each GO term that appears during a GO term annotation import.

Resolving Alternate Identifiers to Database Frames

Each frame object in the database is uniquely identified by an internal identifier known as the frame ID. The BioCyc framework supports annotating frames with alternate identifiers, such as those which are commonly used in literature to refer to genes, proteins, and other biological objects. For example, "PYRUVATE" in EcoCyc has the synonyms alpha-ketopropionic acid, BTS, α-ketopropionic acid, acetylformic acid, pyroracemic acid, 2-oxopropanoic acid, pyruvic acid, 2-oxopropanoate, and 2-oxo-propionic acid. Despite the availability of these alternate identifiers, all queries to the database must resolve to valid frame IDs. A key benefit of CycTools is support for automatically resolving alternate identifiers into frame IDs, removing the need for researchers to perform the conversion manually. Alternate identifiers must already be annotated to the object they identify within the database and must be stored in one of the slots designated as a "name" slot in Pathway Tools. These slots typically include the "accession" slot, "common-name" slot, "synonym" slot, and foreign database identifiers used in the "dblink" slot, but can vary with object type.

During the import process, CycTools attempts to resolve all user provided identifiers into frame IDs. First, CycTools checks if the user provided identifiers match exactly to any existing frame IDs. If all identifiers are determined to be valid frame IDs, no further action is needed and the ID resolution step is skipped. If one or more IDs are not valid frame IDs, CycTools will attempt to resolve them into valid frame IDs using an indexed text search within the database using the "substring-search" method provided by the Pathway Tools API. The substring-search command can find objects with frame IDs that exactly match the search string which match to a substring of any "name" slot. The search term provided by the user must be at least 3 characters with no commas or spaces. This method requires the user to specify the object type to search and the alternate identifiers to be converted to frame IDs. For each identifier in the import file, CycTools requires that the searched term match exactly and entirely to at least one synonym provided by the database for the matching object. Thus, while substring search will match a partial identifier to a frame, CycTools enforces a stricter matching policy by filtering out matches that do not contain complete matches to an alternate identifier. Additionally, CycTools requires that only one such matching object be found in the database. If the search returns only a single frame, that frame's ID is substituted for the searched term. If multiple matches or no match is found, the user is given the option to ignore that data during import, or to cancel the import process altogether.

Create Transcriptional Regulation Frames

Importing novel transcriptional regulatory interactions requires creating regulation frames within the BioCyc database to represent the interaction. Since this import type generates new frames rather than modifying existing ones, the user does not provide frame identifiers with the import data. As a result, no frame ID search is necessary. CycTools instead requests unique sequential identifiers for each new regulation object created. CycTools is not able to recognize if an equivalent regulatory interaction exists in another regulation frame, and therefore relies on the user to ensure that regulatory interactions are not duplicated.

Delete Frames

CycTools implements frame deletion using the Pathway Tools API method "delete-frame-and-dependents". This method detects the object type of the frame which is being deleted and attempts to also delete any frames which depend on the deleted frame. For example, deleting a gene frame will also delete the gene's products, and potentially enzymatic reactions which depend on an enzyme produced by the gene. Regulation frames and history note frames linked to the deleted frame are also deleted.

Preview Changes

Before any permanent modification is made to the database, the user can preview the pending changes to the database. A list shows all frames that will be updated as per the user data. Individual frames can be viewed which will compare the original frame data to the modified data. All changes between the original and modified frames will be highlighted to help the user more easily verify the import. The differences are calculated using a free library called google-diff-match-patch [8]. Highlighting is inferred from the text differences reported by the diff tool.

Commit to Database

After the update is performed, the results of the update can be reviewed. This will provide a log of the successful and failed imports which can be used to verify the success of the import, or to track down problems with the data. Each individual import will be listed as either successful or failed, will be time stamped, and will refer to the original row of data in the spreadsheet which that update represents. Note that it may be possible to have several updates refer to the same row of data. At this point, the database is in a modified but unsaved state. If the user is satisfied with the update, the changes can be permanently saved to the database. Otherwise, the user can undo all changes to the database since the last save. The user will also be given the option of saving the change log to a file.

Import Error Detection

CycTools checks for errors and provides user feedback at several points during the import process. CycTools will directly reject syntax errors such as bad file formats of invalid references to database objects. Illegal database operations on the BioCyc database will cause failed imports in the final commit step, which will be flagged to users so that they can revert the database to an unmodified state. Imports with identifiers which cannot be resolved to existing database objects will be reported to the user as such.

Many errors in data entry are technically valid and thus cannot be differentiated from intentional input. If a slot label is misspelled, for example, CycTools will assume the user intends to create a slot using the misspelled label. The preview step provides users with a frame-by-frame comparison of the database in a modified and an unmodified state. Users are encouraged to browse the anticipated changes in order to detect any data entry errors that would otherwise be valid imports.

Results and Discussion

Use Case: MaizeCyc and CornCyc GO Term Annotation Migration

MaizeCyc [9] and CornCyc [10] are two separate BioCyc databases both based on the *Zea mays* B73 RefGen_v2 gene models [11]. MaizeCyc is developed by Gramene [12] in collaboration with MaizeGDB [13] and CornCyc is developed by Plant Metabolic Network [14] and MaizeGDB [15, 16, 17]. Recent comparison between MaizeCyc and CornCyc revealed annotation differences in data content and quality despite both databases having been based on the same reference sequence [18, 19]. MaizeCyc does not contain alternative splicing information; therefore each gene is only linked to a single gene product. CornCyc does contain alternative splicing information, where gene products linked to alternate splice variants are suffixed with a numerical identifier. It is interesting to note that even though MaizeCyc does not contain alternative splicing information, it still uses the numerical suffix convention for differentiating between alternately spliced proteins.

Recent curation efforts have provided GO term annotations for several proteins in the MaizeCyc database; however CornCyc version 4.0 does not currently contain any GO annotations. Since MaizeCyc and CornCyc both were created using the same sequence data and represent the same biology, the biological functions of MaizeCyc genes should be identical to those of CornCyc genes. In an effort to update the GO term annotations of the maize genome databases and ensure consistency across both databases, the manually curated GO annotations needed to be transferred from MaizeCyc to CornCyc.

All GO term assignments and their annotations were exported from MaizeCyc using a query to the Pathway Tools API. GO term / Annotation pairs with an evidence code beginning with EV-EXP (i.e. experimentally verified annotations) were retained, while all others were removed. This represents the GO term annotations which have been manually verified by curators. Source protein objects were identified by their gene model name (e.g. GRMZM2G136161_P01) with the splice variant suffix attached (i.e. the _P01). This identifier was chosen as it is provided as a synonym in both MaizeCyc and CornCyc, which allows for accurate mapping between objects in both databases. Although MaizeCyc and CornCyc were built using the same gene model set, the internal frame IDs of the protein objects in Pathway Tools were generated with different syntax rules (i.e. most proteins in MaizeCyc begin with GBWI, while the equivalent proteins in CornCyc begin with GDQC).

In order to ensure the most faithful mapping between MaizeCyc and CornCyc proteins, protein identifiers from MaizeCyc were used as query terms in a substring (synonym) search in CornCyc. Exactly matching splice variants provided 179 matches between MaizeCyc and CornCyc as seen in Figure 2.5. While an additional 5 matches can be made between this group of MaizeCyc and CornCyc proteins by relaxing the requirements to allow matches between alternate splice variants, these additional matches were not included in the final import. The remaining 458 gene products from MaizeCyc with EV-EXP annotations do not exist in Corn-

Cyc. The annotation data for the 179 matching protein GO term annotations were inserted into the CornCyc database using the CycTools import feature.

Use Case: Creating Strain-specific EcoCyc Databases

Metabolic engineering projects lead to the generation of genetically unique strains. These altered strains are metabolically similar to the parent strain, but include a small number of modifications such as gene additions, deletions, or regulatory changes. Many novel strains may be created as a result of iterative engineering interventions performed on a parent strain. One possible solution to storing this information is to generate a new BioCyc database that is synchronized to the altered metabolism of the engineered strain. By using the most up-to-date version of EcoCyc and modifying it with information on engineering interventions, a new database is created which more accurately represents the engineered strain. This use case focuses on modifications to the *E. coli* organism performed for the increase of fatty acid production.

E. coli Strains

Of the many strains of *E. coli* that are represented as model organism databases in the BioCyc database collection, EcoCyc has received the most manual curation. It is therefore desirable to leverage annotations from EcoCyc whenever possible while developing new strain databases. The metabolically engineered strains for which strain specific databases were developed in this study, strain ML103 and strain MLC115-1, were described in Liam et. al [20]. The genotype of ML103 is MG1655 ΔfadD. The genotype of MLC115-1 is MG1655 ΔfadD, ΔpoxB, ackA-pta::cmR

New regulatory links were predicted using the GTRNetwork software [21]. These results were derived for the MG1655 network, and so were applied to a copy of the wildtype EcoCyc database rather than the ML103 or MLC115-1 databases.

Copy EcoCyc

It is important to retain as much known information from the parent strain as possible, therefore the first step is to create a clone of the database representing the parent strain. Once the copy has been prepared, further modifications are necessary to align it to the altered metabolism of the engineered strain. In this case, the EcoCyc *E. coli* MG1655 database is downloaded (available free to academic users, requires registration) [22] and a copy is made to represent our strain specific database.

Strain Specific Updates to EcoCyc

Three types of data were added to the base EcoCyc database in order to represent changes in the engineered strain's metabolism. A gene deletion in the strain is represented in EcoCyc by a deletion of the associated gene object and the gene object's functionality. If the gene product is an enzyme, then that protein product is deleted and any reactions it catalyzes have that enzyme association removed from them. If the reaction has no existing enzymes which can catalyze the reaction, then the reaction is also removed. If the gene is a transcription factor, than the transcription factor is removed as well as any regulation objects in which that transcription factor was either a regulator or target. Preprocessing for this database modification simply requires compiling the list of genes to delete. CycTools automatically removes additional objects which are connected to the deleted gene as described above.

A thioesterase with altered specificity added to the strain improves specificity for specific fatty acid chain lengths. This does not represent novel metabolic functionality in the strain, but rather changes relative activities of an existing functionality. Since kinetic information and relative specificities of enzymes is not stored explicitly in current PGDBs, this information is best added to the comments section of the existing enzyme. Preprocessing in this case requires the user to explicitly write out the comment and provide the identifier of the enzyme to be modified.

The final modification made to the base EcoCyc database is the inclusion of novel computationally predicted transcription factor regulation. These regulatory interactions were predicted using GTRNetwork [21]. Transcription factor regulatory interactions in EcoCyc are typically described by a regulation object which describes a transcription factor's regulatory activity of a transcription factor binding site, but can also be described as a direct interaction between the regulating entity and the regulated gene. As the results produced in this computation prediction tool do not provide predicted binding sites, binding site information is not available for import. Preprocessing in this case requires the user to assemble the list of regulator and target interactions.

Each type of modification to the EcoCyc database must be made separately. In this case, the three modifications, gene deletions, thioesterase comment, and predicted regulation, represent three types of modification. Gene deletions are removed from the database by selecting the frame deletion option and loading the list of genes to be deleted. CycTools automatically removes extended links to the provided genes, such as their products and reactions. The thioesterase comment is performed as an update to an existing frame. A file with the comments is loaded and CycTools appends the new comment to the end of any existing comments on the enzyme. Importing novel predicted transcription factor regulation requires creating new regulation frames. This process is performed as two steps internally to CycTools. First, new frames are created using the user provided unique Frame IDs. An import step is then used to load the regulation data into the newly created regulation frames.

Conclusions

Managing and migrating manual annotations in model organism databases are essential to maintaining high-quality biological data. In this work we present a software tool which provides a simple pipeline for the maintenance and transfer of manual annotations within and between BioCyc databases. CycTools improves user control over the import process by providing users with methods to edit slot values or slot-value annotations for any frame in a BioCyc database. CycTools also provides methods which allow users to create transcriptional regulatory frames or to delete frames through the import process.

CycTools provides methods that can make small or large-scale edits to a BioCyc database. Databases using the BioCyc framework typically contain between a few frames and several thousand frames. CycTools is capable of processing and displaying several thousand entries, but is limited to a single object type for each import. This means that CycTools is best suited to making many changes to a BioCyc database of a specific type, rather than making many small changes to various object types.

Tracking the changes made to a BioCyc database is made easier with CycTools. The BioCyc framework provides methods to credit an author or organization for frame edits. CycTools allows users to provide curator information which is stored in the BioCyc framework during the import process. CycTools also provides a change log of actions taken during import in order to assist users in recording changes and identifying problems.

In this manuscript, we have demonstrated the utility of CycTools by transferring GO annotations between two databases representing identical biology but having differing data content. We have also demonstrated the ability of CycTools to make several small scale changes to a database in order to customize the content to represent a non-model organism.

Availability and Requirements

Project name: CycTools

Project home page: https://github.com/jrwalsh/CycTools/

Operating system(s): Any platform supporting Java

Programming language: Java

Other requirements: Java 1.7+, Pathway Tools, JavaCycO

Pathway Tools must be installed and running on a Unix-like server system (due to use of the UnixDomainSocket class) and have the relevant PGDB installed. JavaCycO must be running in server mode on the same server as Pathway Tools. For remote connections, JavacycServer listens over a port connection, so this user selected port must be open to outside traffic. CycTools is written in Java and is thus cross-platform compatible, however Java must be installed on the client machine.

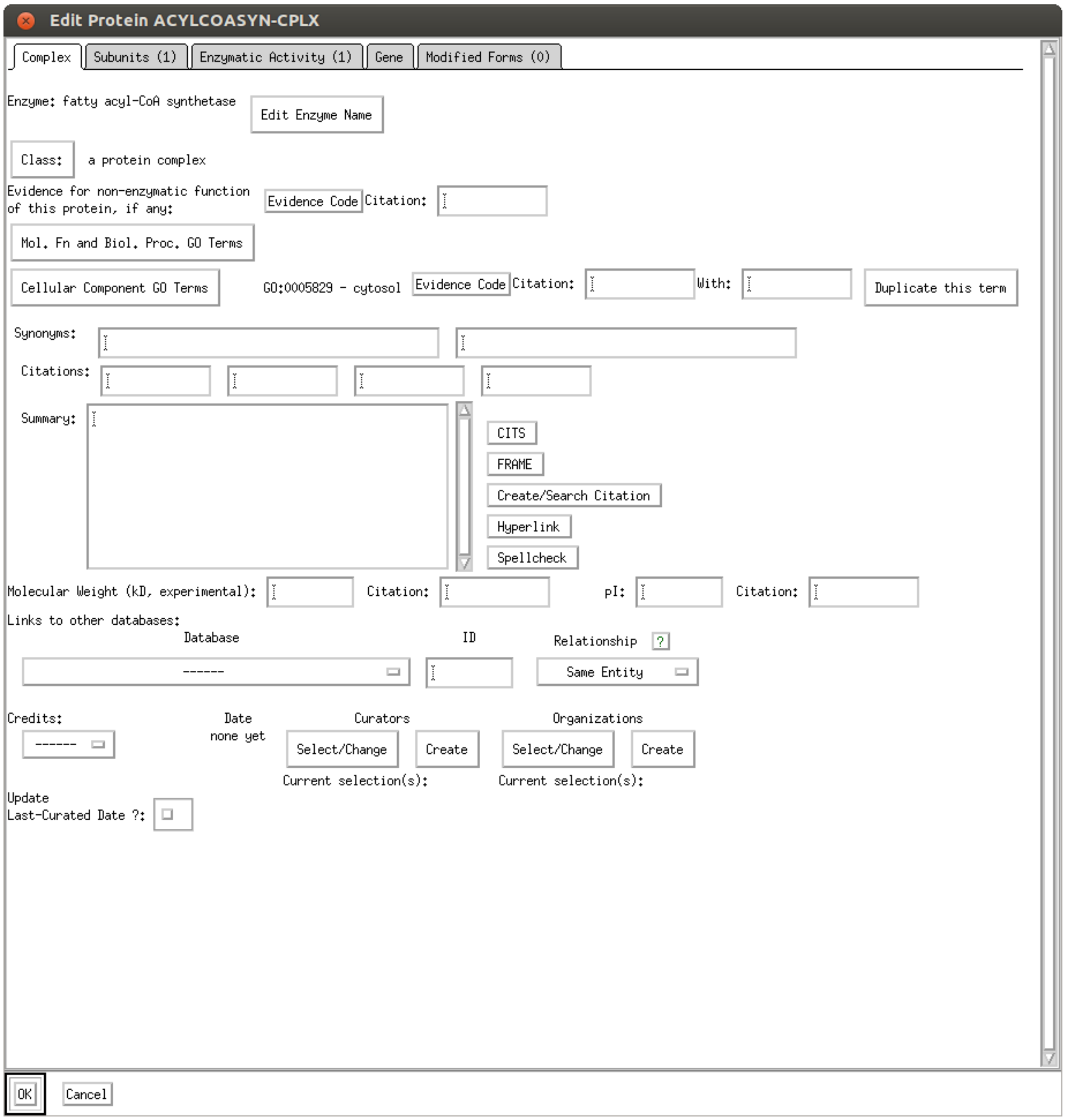
License: GNU GPL **Any restrictions to use by non-academics:** None

Authors' contributions

TZS and JRW conceived, designed, and coordinated the project. JRW developed and documented the software, and drafted the manuscript. JAD provided advice and guidance on the software development and drafting of the manuscript. All authors read and approved the final manuscript.

Acknowledgements

We thank the MaizeGDB team for sharing their insights and expertise. The MaizeGDB group also created a use case for this software and provided user feedback. The material presented here is based upon work supported by the National Science Foundation under Award No. EEC-0813570. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.



Edit Protein ACYLCOASYN-CPLX

Enzyme: fatty acyl-CoA synthetase

Class: a protein complex

Evidence for non-enzymatic function of this protein, if any:

GO:0005829 - cytosol

Molecular Weight (kD, experimental):

Links to other databases:

Database: ------

Relationship: Same Entity

Credits: ------

Date: none yet

Update Last-Curated Date ?: □

Figure 2.1 Screenshot of Pathway Tools Protein Editor. Editing database objects through the Pathway Tools software editors is done be entering information into forms which describe information specific to the type of object being edited.

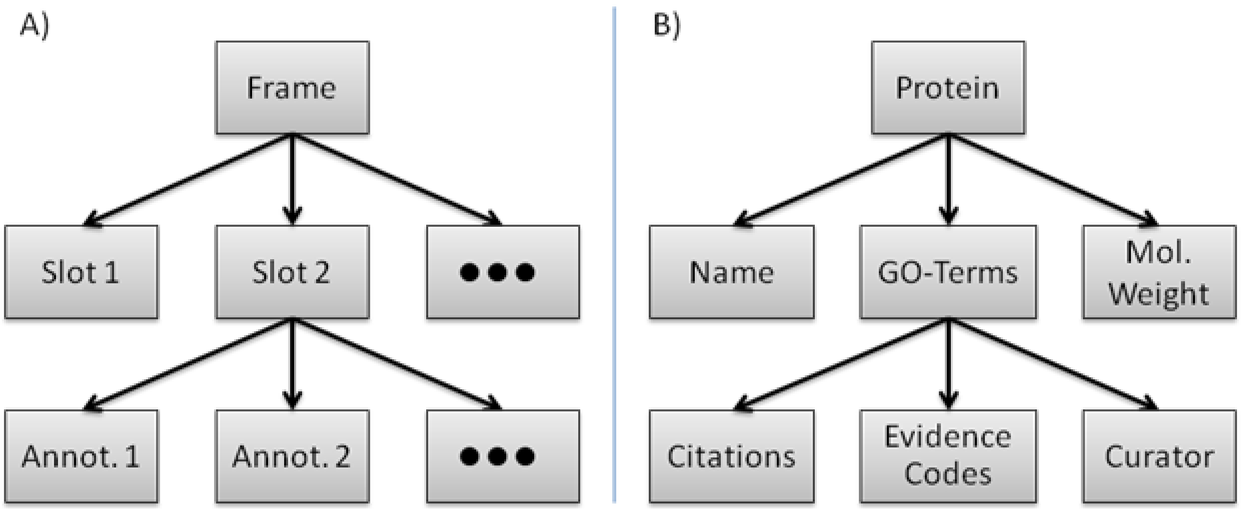


Figure 2.2 Structure of Frames in a PGDB. **A)** Frames describe objects in the database. Slots contain information about the frame object, and annotations contain meta-information about slot information. **B)** A protein is represented by a frame in the database. Examples of slots which describe the protein include the protein name, molecular weight, and GO-Term assignments. Annotations of the GO term information include citations, evidence codes, and information on who curated this GO term assignment.

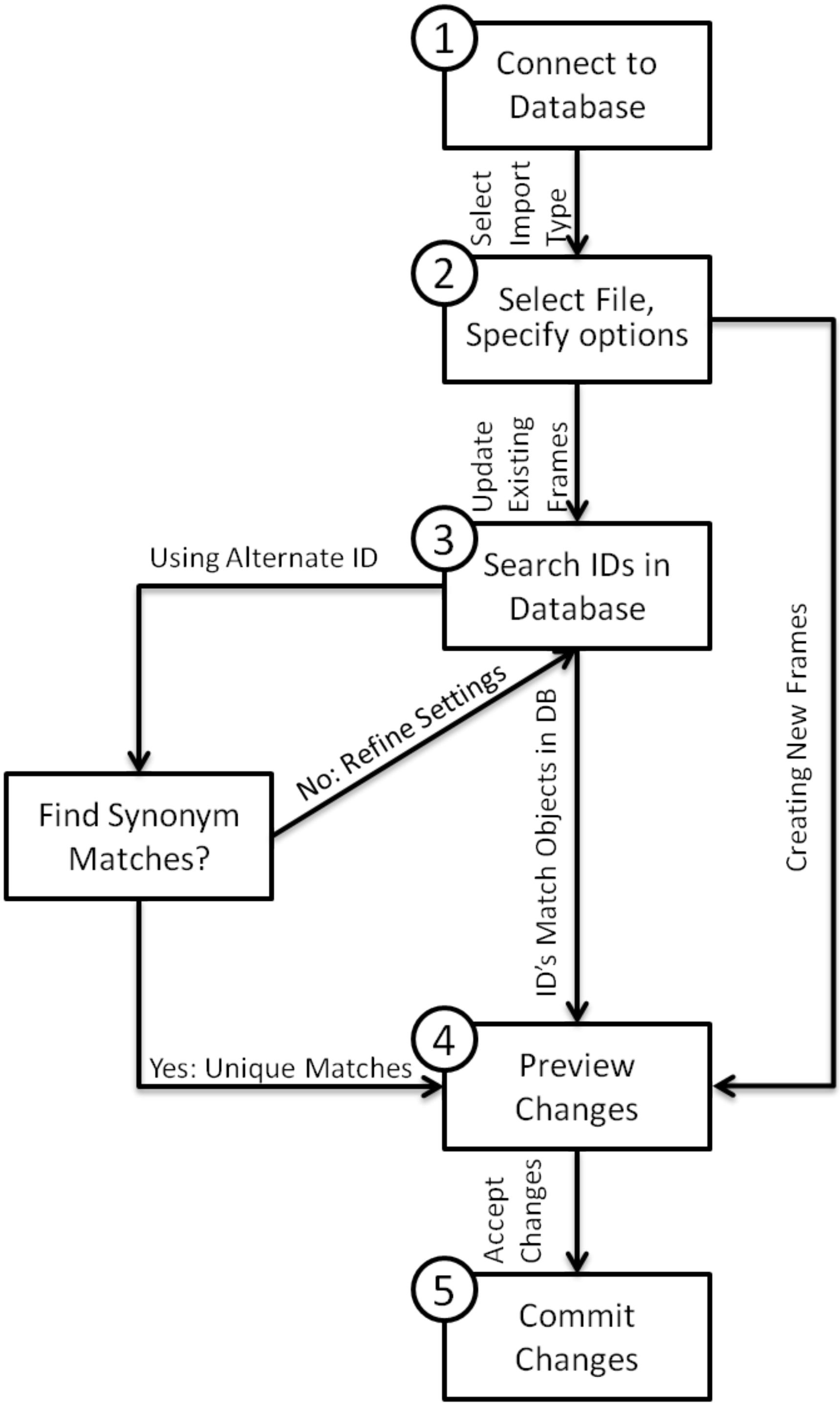
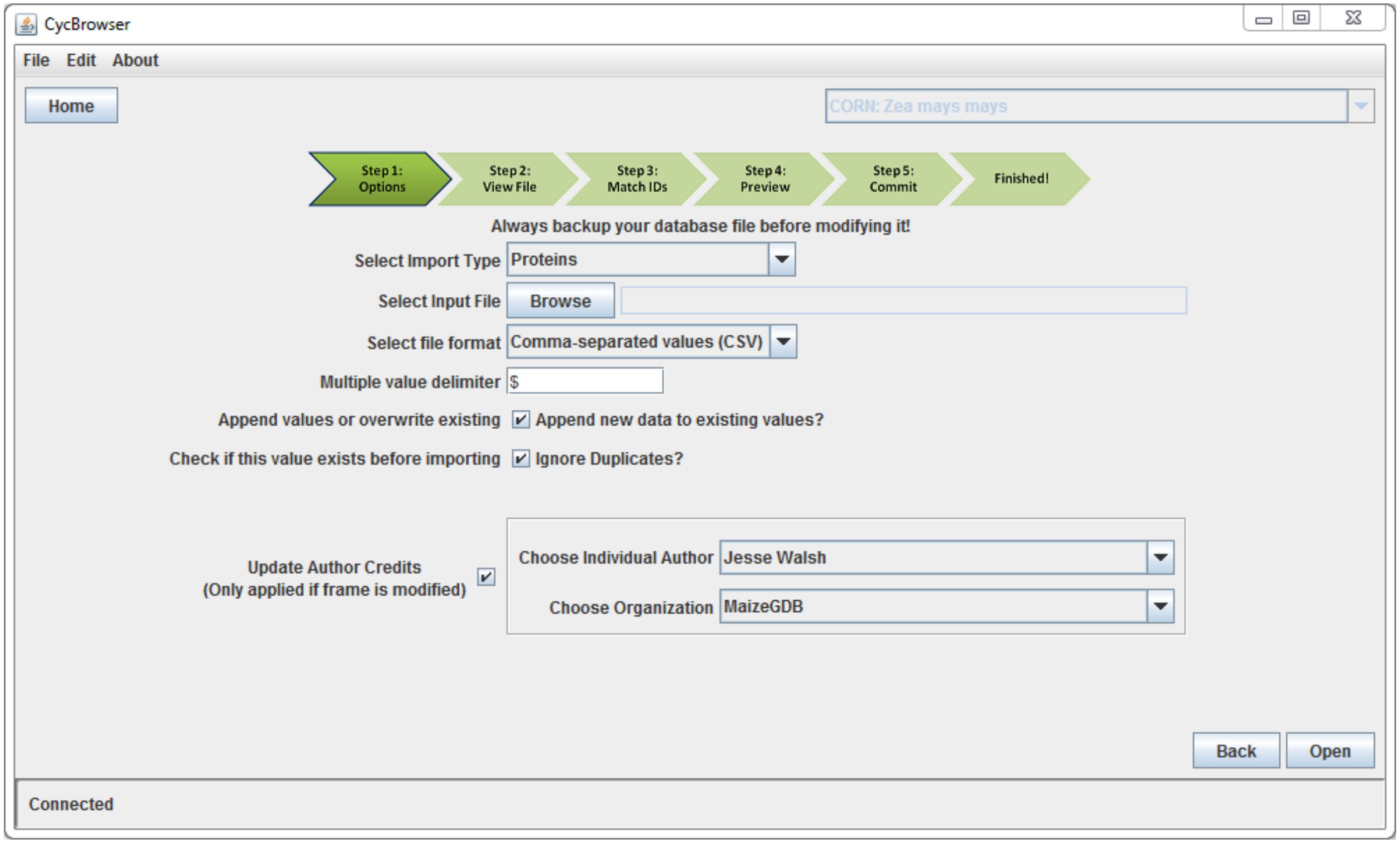


Figure 2.3 Import Process Diagram. Synonym based search automatically occurs if import file does not contain Frame ID's. Only unique matches to frame IDs are allowed in order to prevent ambiguity in the import process.



Always backup your database file before modifying it!

Select Import Type: Proteins

Select Input File

Select file format: Comma-separated values (CSV)

Multiple value delimiter: $

Append values or overwrite existing: ☑ Append new data to existing values?

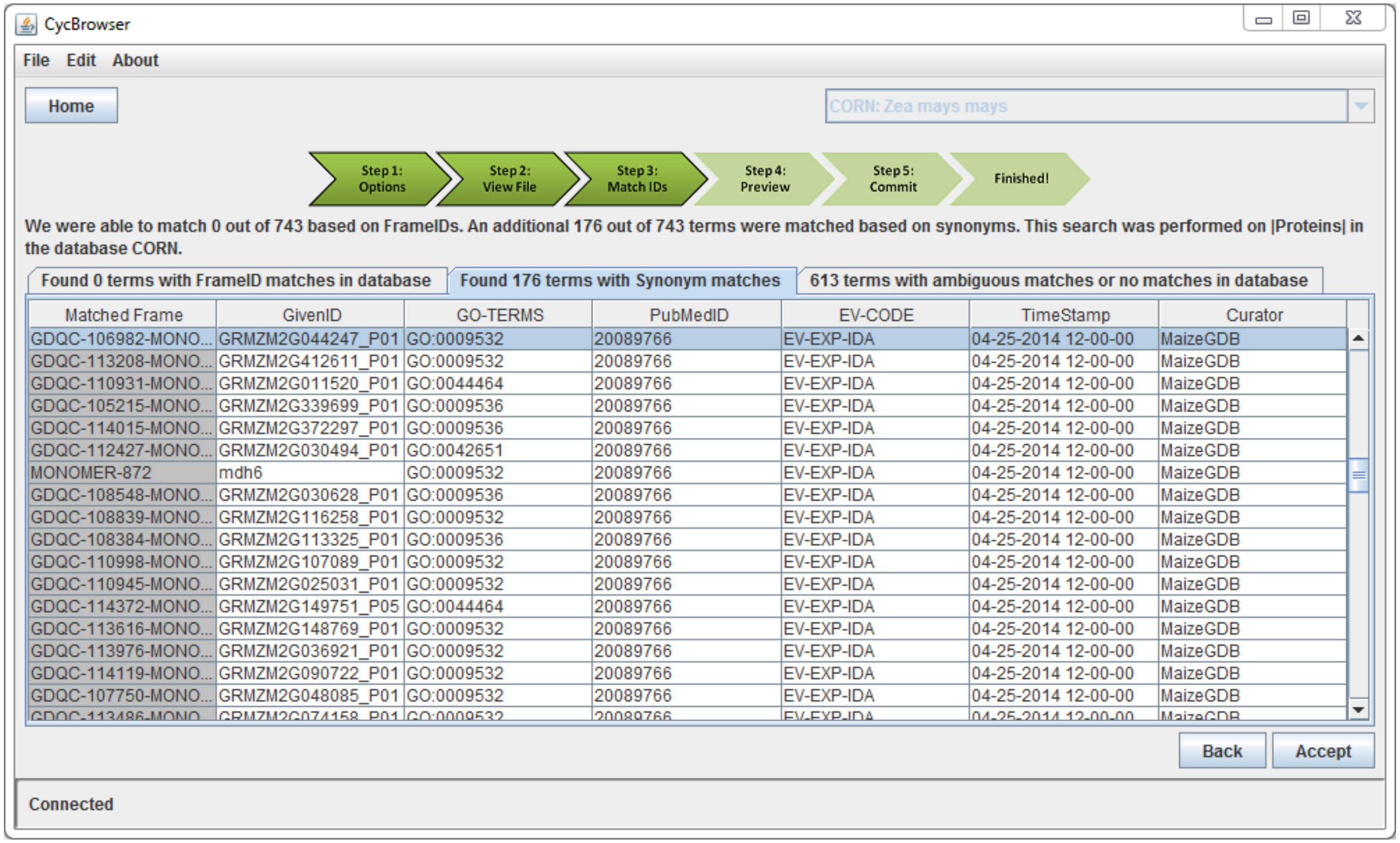
Check if this value exists before importing: ☑ Ignore Duplicates?

Update Author Credits (Only applied if frame is modified) ☑

Choose Individual Author: Jesse Walsh

Choose Organization: MaizeGDB

Figure 2.4 CycTools Import Screen. CycTools provides a multi-step process for importing user data. Several options are available for users to interact with existing data. Users can also specify an author or organization to assign credit for the revision of a database frame object.



We were able to match 0 out of 743 based on FrameIDs. An additional 176 out of 743 terms were matched based on synonyms. This search was performed on |Proteins| in the database CORN.

Found 0 terms with FrameID matches in database | Found 176 terms with Synonym matches | 613 terms with ambiguous matches or no matches in database

Matched Frame	GivenID	GO-TERMS	PubMedID	EV-CODE	TimeStamp	Curator
GDQC-106982-MONO...	GRMZM2G044247_P01	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-113208-MONO...	GRMZM2G412611_P01	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-110931-MONO...	GRMZM2G011520_P01	GO:0044464	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-105215-MONO...	GRMZM2G339699_P01	GO:0009536	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-114015-MONO...	GRMZM2G372297_P01	GO:0009536	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-112427-MONO...	GRMZM2G030494_P01	GO:0042651	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
MONOMER-872	mdh6	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-108548-MONO...	GRMZM2G030628_P01	GO:0009536	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-108839-MONO...	GRMZM2G116258_P01	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-108384-MONO...	GRMZM2G113325_P01	GO:0009536	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-110998-MONO...	GRMZM2G107089_P01	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-110945-MONO...	GRMZM2G025031_P01	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-114372-MONO...	GRMZM2G149751_P05	GO:0044464	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-113616-MONO...	GRMZM2G148769_P01	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-113976-MONO...	GRMZM2G036921_P01	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-114119-MONO...	GRMZM2G090722_P01	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-107750-MONO...	GRMZM2G048085_P01	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB
GDQC-113486-MONO	GRMZM2G074158_P01	GO:0009532	20089766	EV-EXP-IDA	04-25-2014 12-00-00	MaizeGDB

Figure 2.5 GO Term Annotation Import into CornCyc. GO term annotations obtained from MaizeCyc are imported into CornCyc. User provided gene model identifiers are resolved to database frame IDs before import.

Bibliography

[1] Karp, P.D., Ouzounis, C.A., Moore-Kochlacs, C., Goldovsky, L., Kaipa, P., Ahrn, D., Tsoka, S., Darzentas, N., Kunin, V., Lpez-Bigas, N.: Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. Nucleic Acids Research **33**(19), 6083–6089 (2005). PMID: 16246909

[2] Schnoes, A.M., Brown, S.D., Dodevski, I., Babbitt, P.C.: Annotation error in public databases: Misannotation of molecular function in enzyme superfamilies. PLoS Comput Biol **5**(12), 1000605 (2009)

[3] Group, T.B.: Guide to the BioCyc Database Collection (2014). http://biocyc.org/BioCycUserGuide.shtml Accessed 2014-05-08

[4] Paley, S.M., Latendresse, M., Karp, P.D.: Regulatory network operations in the pathway tools software. BMC Bioinformatics **13**(1), 243 (2012). PMID: 22998532

[5] Group, T.B.: Ocelot User's Guide (2014). http://www.ai.sri.com/pkarp/ocelot/ Accessed 2014-05-08

[6] Krummenacker, M., Paley, S., Mueller, L., Yan, T., Karp, P.D.: Querying and computing with BioCyc databases. Bioinformatics **21**(16), 3454–3455 (2005)

[7] Van Hemert, J.L., Dickerson, J.A.: PathwayAccess: CellDesigner plugins for pathway databases. Bioinformatics **26**(18), 2345–2346 (2010)

[8] Group, T.D.: google-diff-match-patch - Diff, Match and Patch libraries for Plain Text - Google Project Hosting (2014). http://code.google.com/p/google-diff-match-patch/ Accessed 2014-05-08

[9] Monaco, M.K., Sen, T.Z., Dharmawardhana, P.D., Ren, L., Schaeffer, M., Naithani, S., Amarasinghe, V., Thomason, J., Harper, L., Gardiner, J., Cannon, E.K., Lawrence, C.J., Ware, D., Jaiswal, P.: Maize metabolic network construction and transcriptome analysis. The Plant Genome **6**(1), 0 (2013)

[10] Chae, L., Lee, I., Shin, J., Rhee, S.Y.: Towards understanding how molecular networks evolve in plants. Current Opinion in Plant Biology **15**(2), 177–184 (2012)

[11] Schnable, P.S., Ware, D., Fulton, R.S., Stein, J.C., Wei, F., Pasternak, S., Liang, C., Zhang, J., Fulton, L., Graves, T.A., Minx, P., Reily, A.D., Courtney, L., Kruchowski, S.S., Tomlinson, C., Strong, C., Delehaunty, K., Fronick, C., Courtney, B., Rock, S.M., Belter, E., Du, F., Kim, K., Abbott, R.M., Cotton, M., Levy, A., Marchetto, P., Ochoa, K., Jackson, S.M., Gillam, B., Chen, W., Yan, L., Higginbotham, J., Cardenas, M., Waligorski, J., Applebaum, E., Phelps, L., Falcone, J., Kanchi, K., Thane, T., Scimone, A., Thane, N., Henke, J., Wang, T., Ruppert, J., Shah, N., Rotter, K., Hodges, J., Ingenthron, E., Cordes, M., Kohlberg, S., Sgro, J., Delgado, B., Mead, K., Chinwalla, A., Leonard, S., Crouse, K., Collura, K., Kudrna, D., Currie, J., He, R., Angelova, A., Rajasekar, S., Mueller, T., Lomeli, R., Scara, G., Ko, A., Delaney, K., Wissotski, M., Lopez, G., Campos, D., Braidotti, M., Ashley, E., Golser, W., Kim, H., Lee, S., Lin, J., Dujmic, Z., Kim, W., Talag, J., Zuccolo, A., Fan, C., Sebastian, A., Kramer, M., Spiegel, L., Nascimento, L., Zutavern, T., Miller, B., Ambroise, C., Muller, S., Spooner, W., Narechania, A., Ren, L., Wei, S., Kumari, S., Faga, B., Levy, M.J., McMahan, L., Buren, P.V., Vaughn, M.W., Ying, K., Yeh, C.-T., Emrich, S.J., Jia, Y., Kalyanaraman, A., Hsia, A.-P., Barbazuk, W.B., Baucom, R.S., Brutnell, T.P., Carpita, N.C., Chaparro, C., Chia, J.-M., Deragon, J.-M., Estill, J.C., Fu, Y., Jeddeloh, J.A., Han, Y., Lee, H., Li, P., Lisch, D.R., Liu, S., Liu, Z., Nagel, D.H., McCann, M.C., SanMiguel, P., Myers, A.M., Nettleton, D., Nguyen, J., Penning, B.W., Ponnala, L., Schneider, K.L., Schwartz, D.C., Sharma, A., Soderlund, C., Springer, N.M., Sun, Q., Wang, H., Waterman, M., Westerman, R., Wolfgruber, T.K., Yang, L., Yu, Y., Zhang, L., Zhou, S., Zhu, Q., Bennetzen, J.L., Dawe, R.K., Jiang, J., Jiang, N., Presting, G.G., Wessler, S.R., Aluru, S., Martienssen, R.A., Clifton, S.W., Mc-

Combie, W.R., Wing, R.A., Wilson, R.K.: The b73 maize genome: Complexity, diversity, and dynamics. Science **326**(5956), 1112–1115 (2009). PMID: 19965430

[12] Team, T.G.: MaizeCyc Database Home, Metabolic Pathways in Maize or Corn (2014). http://pathway.gramene.org/maizecyc.html Accessed 2014-06-04

[13] Team, T.M.: MaizeCyc Database Home, Metabolic Pathways in Maize (2014). http://maizecyc.maizegdb.org/ Accessed 2014-05-08

[14] Team, T.P.: Summary of Zea mays, Subspecies mays, version 4.0.1 (2014). http://pmn.plantcyc.org/organism-summary?object=CORN Accessed 2014-05-08

[15] Team, T.M.: CornCyc Database Home, Metabolic Pathways in Maize (2014). http://corncyc.maizegdb.org/ Accessed 2014-05-08

[16] Sen, T.Z., Andorf, C.M., Schaeffer, M.L., Harper, L.C., Sparks, M.E., Duvick, J., Brendel, V.P., Cannon, E., Campbell, D.A., Lawrence, C.J.: MaizeGDB becomes 'sequence-centric'. Database: The Journal of Biological Databases and Curation **2009** (2009)

[17] Lawrence, C.J., Harper, L.C., Schaeffer, M.L., Sen, T.Z., Seigfried, T.E., Campbell, D.A.: MaizeGDB: the maize model organism database for basic, translational, and applied research. International Journal of Plant Genomics **2008**, 496957 (2008)

[18] Team, T.M.: Metabolic Pathways at MaizeGDB (2014). http://alpha.maizegdb.org/metabolic_pathways/compare Accessed 2014-05-08

[19] Sen, T.Z.: unpublished results (2014)

[20] Royce, L.A., Liu, P., Stebbins, M.J., Hanson, B.C., Jarboe, L.R.: The damaging effects of short chain fatty acids on escherichia coli membranes. Applied Microbiology and Biotechnology **97**(18), 8317–8327 (2013)

[21] Fu, Y., Jarboe, L.R., Dickerson, J.A.: Reconstructing genome-wide regulatory network of e. coli using transcriptome data and predicted transcription factor activities. BMC Bioinformatics **12**(1), 233 (2011)

[22] Group, T.B.: Pathway Tools Download (2014). http://biocyc.org/download-bundle.shtml Accessed 2014-05-08

3. THE QUALITY OF METABOLIC PATHWAY RESOURCES DEPENDS ON INITIAL ENZYMATIC FUNCTIONAL ASSIGNMENTS AND LEVEL OF MANUAL CURATION: A CASE FOR MAIZE

A paper to be submitted to Plant Physiology

Jesse R. Walsh, Mary L. Schaeffer, Peifen Zhang, Seung Y. Rhee, Julie A. Dickerson, and Taner Z. Sen

Abstract

As metabolic pathway resources become more commonly available, researchers have unprecedented access to information about their organism of interest. Despite efforts to ensure consistency between various resources, information content and quality can vary widely. Two maize metabolic pathway resources for the B73 inbred line, CornCyc4.0 and MaizeCyc2.2, are based on the same gene model set and were developed using Pathway Tools software. These resources differ in their initial enzymatic function assignments and in the extent of manual curation. We present an in-depth comparison between CornCyc and MaizeCyc to demonstrate the effect of initial computational enzymatic function assignments on the final quality and content of metabolic pathway resources.

MaizeCyc contains over twice as many annotated genes and more proteins than CornCyc. CornCyc contains on average 1.6 transcripts per gene, while MaizeCyc contains almost no alternate splicing. MaizeCyc does not match CornCyc's breadth in representing the metabolic domain. MaizeCyc has fewer compounds, reactions, and pathways than CornCyc. CornCyc's computational predictions are more accurate than those in MaizeCyc when compared

to experimentally determined function assignments, demonstrating the relative strength of the enzymatic function assignment pipeline used to generate CornCyc.

Our results show that the quality of initial enzymatic function assignments primarily determines the quality of the final metabolic pathway resource. Therefore, biologists should pay close attention to the methods and information sources used to develop a metabolic pathway resource to gauge the utility of using such functional assignments to construct hypotheses for experimental studies.

Introduction

Developing a metabolic pathway resource involves many steps. These steps can be described as follows: Given a genome assembly and a gene model set, translated protein sequences are fed into a computational pipeline. Enzymes are then predicted and assigned a functional category, usually based on Gene Ontology (GO) [1] terms or Enzyme Commission (EC) [2] numbers. After the initial enzymatic function assignments are made, enzymes are then mapped to a reference metabolic pathway database to create an initial metabolic pathway resource. Finalizing a pathway resource requires manual curation to improve the accuracy of the final metabolic representation.

A wide-range of computational methods can be applied at each step of developing a metabolic pathway resource. This variance makes a comparison of metabolic pathway resources challenging. The problems that complicate comparison between heterogeneous databases have long been recognized [3], and several attempts have been made to homogenize data from different sources [4, 5]. Studies seeking to compare data content between resources [6] describe many of the challenges of matching biological data in order to assess overlap. Non-standard chemical naming conventions, difficulty matching stereo-chemistry and protonation, as well as defining pathway boundaries and managing gene variants all create challenges for comparing metabolic pathway resources.

For maize, two metabolic network resources are available, both of which are based on the B73 RefGen_v2 genome assembly/gene model set [7] and used the Pathway Tools software [8] to map enzymes onto reactions and pathways. This provides a unique opportunity to explore the effect of the initial enzymatic function assignment pipeline on the final metabolic pathway resource.

CornCyc4.0 [9] was developed using the Ensemble Enzyme Prediction Pipeline [10] created by Plant Metabolic Network (PMN) [11] in collaboration with MaizeGDB (`http://www.maizegdb.org`) [12, 13]. MaizeCyc2.2 [14] was developed based on the Ensembl XRef pipeline [15, 16] in collaboration between two database projects, Gramene (`http://www.gramene.org`) and MaizeGDB. The term "Ensemble" in the CornCyc pipeline refers to integration of methods, whereas "Ensembl" in the MaizeCyc pipeline refers to the collaborative project between the European Bioinformatics Institute and the Wellcome Trust Sanger Institute.

In order to gain insight into the strengths of each resource based on initial enzymatic function assignments, we compared the data content and accuracy of CornCyc and MaizeCyc. To accomplish this goal, we calculated the overlap between the resources in multiple categories (gene, protein, compound, reaction, and pathway) as well as compared the resources against experimentally-assigned enzymatic functions.

Methods

Resource Preparation and Access

We compared CornCyc version 4.0 with MaizeCyc version 2.2 hosted within Pathway Tools 17.0 [8]. Both resources are based on the B73 RefGen_v2 reference genome assembly and the filtered gene set (FGS) [7]. Throughout the text, we refer to CornCyc4.0 as CornCyc and MaizeCyc2.2 as MaizeCyc unless otherwise specified. Although the v2 assembly of the maize genome sequence is not as recent as the v3 assembly, MaizeCyc was only available for v2, which drove our decision to use a previous assembly. The current version CornCyc6.0 uses the more recent v3 assembly. We used an older version of CornCyc as it used the v2 assembly. Also, while MaizeCyc was developed using Pathway Tools 17.0, CornCyc was developed using Pathway Tools 16.5. We upgraded CornCyc to Pathway Tools 17.0 using PathoLogic built-in upgrade process. Our comparison between the original and the upgraded CornCyc versions shows that the change is negligible: for example, no reactions and only 0.4% of pathways are affected by the upgrade for the purpose of comparing CornCyc and MaizeCyc. The decision to use Pathway Tools 17.0 was made in order to compare both resources. All data extraction queries to the CornCyc and MaizeCyc resources were made using the JavaCycO libraries [17] and the Pathway Tools Application Program Interface (API). Details of the methods used to extract and compare the data from CornCyc and MaizeCyc are available in Appendix B.

CornCyc Annotation Pipeline

CornCyc was developed based on the Ensemble Enzyme Prediction Pipeline (E2P2) [10]. E2P2 uses an average weighted integration algorithm based on results from individual classifiers such as BLAST [18], CatFam [19], and Priam [20]. The ensemble algorithm relies on an average weighted integration scheme where the weight of each predicted model was determined by a 5-by-3 nested cross-validation routine. For CornCyc version 4.0, E2P2 version 2.1 (https://dpb.carnegiescience.edu/labs/rhee-lab/software) was used with BLAST's e-value cutoff set to be $<= 1e-30$. The training of E2P2 and the reference databases used in the annotation process are based on the Reference Protein Sequence Dataset (RPSD) version 2.0 included in the E2P2 v2.1 package. RPSD contains protein sequences with experiment support of existence compiled from SwissProt [21], MetaCyc [22], and BRENDA [23].

After the initial database generation, CornCyc was further modified using the SAVI pipeline [11], which categorizes the initially predicted pathways to be retained, deleted, or manually reviewed based on a set of rules developed as a part of curation process. SAVI also detects missing pathways. The SAVI program uses six curated pathway library files to enable semi-automated changes to a predicted pathway database (http://www.plantcyc.org/about/savi_pipeline.faces). All pathway library files used in validating and refining CornCyc 4.0 are available online at: ftp://ftp.plantcyc.org/Pathways/SAVI_validation_lists/SAVI_validation_lists_archive/SAVI_lists_pmn8_july_2013/

MaizeCyc Annotation Pipeline

The development pipeline for MaizeCyc was described in detail previously [14]. MaizeCyc is based on the B73 RefGen_v2 filtered gene set. The pipeline uses the "canonical transcript" with the longest open reading frame for functional annotation based on scores derived from the Ensembl XRef pipeline [16] following protein sequence alignment to UniProt [21]. Additional sources of enzymatic function annotations include classical maize genes [24], coordinates and cross-references from Maizesequence.org (now folded into Gramene), MaizeGDB (locus names/ synonyms, molecular function, etc.) [12, 13], UniProtKB/SwissProt [21] (functional descriptions and EC assignments), Gene Ontology [1] (mol. function, biol. process, cellular location), and proteomics-supported gene annotations (e.g., cellular location). Reactions and pathways were computationally inferred using the Pathologic component of Pathway Tools [8].

Availability and Requirements

The software used to query CornCyc and MaizeCyc is available as an executable Java program at https://github.com/jrwalsh/CornCompare. Pathway Tools must be installed and running on a Unix-like server system (due to use of the UnixDomainSocket class) and have CornCyc and MaizeCyc installed. JavaCycO must be running in server mode on the same server as Pathway Tools. This software was written in Java and is thus cross-platform compatible when Java is installed on the client machine.

Results and Discussion

Validation of Enzymatic Function Assignments Against Experiments

To compare the prediction accuracy between these databases, we extracted 197 experimentally determined enzyme annotations for maize from UniProt [21]. Then we matched these proteins to the B73 RefGen_v2 gene models using BLAST based on a sequence identity cutoff of 96% and coverage of 90 which reduced the number of maize enzymes to 177. We then calculated the precision, recall, and F-score values by comparing computational predictions of EC numbers against experimentally determined assignments. The formulas of the performance measures are provided below.

We used the following definitions for our performance classifications: 1) true positive (TP) is when a predicted function of an enzyme matches an experimentally determined function category for that enzyme. 2) False positive (FP) is when a predicted function does not match any experimentally determined function category for that enzyme. Finally 3) false negative (FN) is when a function category is an experimentally determined but is not predicted by the annotation algorithm. The fourth category, true negative (TN), is a quantity that is difficult to capture, as it means that for a given enzyme no prediction is made for a functional category that is also ruled out experimentally. Precision, recall, and F-score only uses TP, FP, and FN classifications. A summary of the results is shown in Table 3.1.

We used the following expressions for analysis: $Precision = TP/(TP + FP), recall = TP/(TP+FN)$. Precision is a ratio of correctly predicted classes among all the predictions, and recall is a ratio of correctly predicted classes among all the possible correct classes. F-score is a combination of these two measures and provides a single measure for comparing the performance of two sets of predictions. F-score is defined as $(precision * recall)/(precision + recall)$.

CornCyc performs better than MaizeCyc, as demonstrated by higher precision (0.88 versus 0.79), recall (0.91 versus 0.24), and F-score (0.90 versus 0.38) (Figure 3.1). CornCyc's performance originates from the much higher number of true positives and much lower number of false negatives in its predictions. For biologists, what this means is that when they find an annotation in CornCyc, it is more likely to be correct than it is in MaizeCyc.

Comparison of Data Overlap

The comparison between genes in CornCyc and MaizeCyc in Figure 3.2 shows a large divergence between the two resources, despite the fact that both were developed based on the same gene model set. Part of the reason for this is that the scope of MaizeCyc includes all genes in the maize genome, while the scope of CornCyc includes only enzyme-coding genes. In order to draw a useful comparison between the gene content in CornCyc and MaizeCyc, we only considered genes associated with a form of annotation. Specifically, we define a gene to have annotation if it is either assigned at least one GO term or is associated with a protein that catalyzes at least one reaction without any filtering by evidence codes. In CornCyc, only 2.0% of the 9,142 genes have GO term annotations, but 99.7% are mapped to at least one reaction.

In MaizeCyc, 53% of the 39,654 genes have GO term annotations while only 8.1% are mapped to at least one reaction (Figure 3.2).

Only proteins that mapped to at least one reaction were included in the analyses. An additional step in the matching process was needed to handle a group of approximately 5,800 proteins in MaizeCyc that did not include a gene model name as a synonym. These proteins are annotated with non-specific and non-unique names (i.e., 325 such proteins are named "Nucleoside-triphosphatase"). For this group, we used the gene model name of the associated transcript rather than the name of the protein to perform the comparison. Figure 3.2 shows a large increase in the number of proteins unique to CornCyc compared to the number of unique genes in CornCyc. This is due in large part to the fact that this version of CornCyc represents alternative splice variants and contains on average 1.6 splice variants per gene. In contrast, MaizeCyc includes very few alternatively spliced variants.

While 1,686 reactions were found in both CornCyc and MaizeCyc, CornCyc contains 1,275 reactions not present in MaizeCyc and MaizeCyc contains 591 reactions not present in CornCyc (Figure 3.2). In order to determine if the differences in reaction content reflect differences in coverage of reaction space, we compared the distribution of Enzyme Commission (EC) categories for the reactions in each resource. Reactions were assigned to EC categories using their top-level EC class. We compared the total reaction content of CornCyc and MaizeCyc to the portion of reactions unique to CornCyc and MaizeCyc, as well as the total reaction content of BRENDA [23] and MetaCyc [22] (Figure 3.3). MetaCyc contains the source reactions from which CornCyc and MaizeCyc imported their reaction information, while BRENDA contains a comprehensive source of enzyme information derived from literature.

Table 3.2 shows the distribution and overlap of reactions by top-level EC category for CornCyc and MaizeCyc. A total of 256 reactions in CornCyc and 185 reactions in MaizeCyc were not assigned an EC number. Reactions might be missing an EC number in three cases: 1) the reaction is pending review by the EC commission, 2) the reaction is hypothetical without an experimentally characterized enzyme activity, or 3) if the reaction is not associated with an enzyme such as the case for some transport reactions. Of the reactions without EC assignments, 21 were classified as transport reactions in CornCyc and 94 were classified as transport reactions in MaizeCyc. CornCyc has more unique reactions than MaizeCyc in all EC categories. Figure 3.3 shows the frequency of reactions in each category for each resource. Comparing the reactions unique to CornCyc and MaizeCyc reveals that CornCyc has stronger representation than MaizeCyc in each category except lyases.

We considered only small, non-elemental molecules (i.e., no proteins, no DNA/RNA, etc.). Since compounds are imported into the CornCyc and MaizeCyc from MetaCyc, we do not expect them to be intrinsically unique in one resource except when the two resources contain reactions catalyzing compounds unique to those reactions. As expected, CornCyc contains significantly more small-molecule compounds than MaizeCyc (Figure 3.2), providing a greater coverage of the compound space.

CornCyc and MaizeCyc have 280 pathways in common (Figure 3.2). There are 167 and 148 pathways unique to CornCyc and MaizeCyc, respectively. In some cases, difference in pathways seems to originate from updates to the version of MetaCyc that was used. Some examples are: 1) benzoxazinoids biosynthesis, a pathway unique to maize and a few other species [25] (the pathway is absent in MaizeCyc), 2) Indole-3-acetate biosynthesis I based on new evidence about major 2-step pathway for auxin biosynthesis [26] (the pathway variant is absent from MaizeCyc), and 3) alternate C4-photosynthesis pathways: there are two C-4 variants in CornCyc, but only one variant in MaizeCyc.

Some of the differences are caused by variation in functionally similar pathways. A pathway variant might use different co-factors, enzymes, or reactions. Which pathway variant was selected for inclusion in CornCyc and MaizeCyc can be partly attributed to the pathway inference step performed by Pathologic in the CornCyc and MaizeCyc pipelines. Pathologic selects pathways to import from MetaCyc based on the enzyme annotations available within the database. In certain cases, BioCyc resources created from different versions of MetaCyc can be significantly different from one another, which makes pathway variants comparison challenging. For example, the CornCyc pathway Nonaprenyl Diphosphate Biosynthesis I and MaizeCyc pathway Nonaprenyl Diphosphate Biosynthesis III have sharp contrasts in pathway structure. Even a similar pathway structure can lead to different biological and evolutionary implications, such as the C-4 pathway example shown in 3.4. Notwithstanding the gene and enzyme differences, which are overlaid next to their reactions the diagram, the pathways themselves have almost the same metabolic structure. The only difference in the pathway structure is that CornCyc breaks the reaction

$$CO_2 + H_2O + \text{phosphoenopyruvate} \longrightarrow \text{H} + \text{phosphate} + \text{oxaloacetate} \tag{3.1}$$

into two steps, treating the $CO_2 + H_2O \longrightarrow$ hydrogencarbonate part as a separate reaction. From an evolutionary perspective, this is highly significant, since the lack of this step in MaizeCyc implies that the enzyme in MaizeCyc can handle the entire conversion, while CornCyc has two enzymes perform the two steps separately. In the current version of MetaCyc, the version of the pathway present in MaizeCyc is removed, and only the version included in CornCyc is available. Therefore, biologists will benefit from knowing which MetaCyc version was used to create their metabolic resource of interest.

The Level and Quality of Manual Curation Differentiate Metabolic Databases

Manual curation is a powerful approach for ensuring consistency and accuracy of a database. Unfortunately, the time-consuming and expensive nature of curation means that only limited parts of a data resource will receive manual review. In the case of CornCyc and MaizeCyc, their content was populated with computationally predicted annotations during their creation using their respective annotation pipelines. This content is then reviewed in an ongoing curation effort to integrate literature-supported experimental annotation into the metabolic resources.

One area of MaizeCyc that has received considerable manual curation effort is Gene Ontology (GO) annotations. Because GO annotations are important for researchers interested in gene function, we previously developed a tool to migrate GO annotations between Pathway Tools-based metabolic databases [27]. The tool is especially important for preserving valuable manual GO curations between different versions of the same metabolic pathway resources. Previous work reported 789 experimentally verified GO assignments in MaizeCyc, of which 179 were manually transferred to CornCyc by using this tool [27].

The extent of manual curation of reactions and pathways differs for MaizeCyc and for CornCyc. A review of these two resources shows that 91 pathways contain some experimental evidence in CornCyc as opposed to 39 in MaizeCyc. Similarly, there are 120 enzyme-reaction associations in CornCyc that contain experimental evidence. In contrast, MaizeCyc has only 20. An enzyme-reaction association is the assignment of a particular enzyme to a reaction, where one enzyme can potentially catalyze multiple reactions. A large part of curation by MaizeGDB has been to identify gene products confirmed to participate in a pathway, and with a focus on selected pathways involved in hormone metabolism and photosynthesis.

Conclusions

The availability of genome-wide metabolic pathway resources provides a systems-level view of the chemical interactions in a cell, which creates phenotypes of interest. When a metabolic pathway resource is developed and made publicly available, scientists can then construct a network of interactions around their enzymes of interest, and build further hypotheses based on the annotations assigned to the genes and proteins. For example, when an enzyme of interest is discovered to be differentially expressed and hypothesized to play a critical role in the cellular processes, the next step is to gather its functional annotations from several database resources for further analyses. Therefore, it is highly desirable for a metabolic pathway resource to have annotations for larger numbers of enzymes. A higher coverage of the genome-wide enzyme space, however, does not automatically translate into a higher accuracy of prediction for those annotations. Most of these annotations are generated through computational pipelines that involve multiple processing steps, and each step can contribute to the final quality of a metabolic pathway resource. A larger number of functional assignments can indeed provide a higher number of correct assignments (i.e., true positives), but it can also introduce a higher number of wrong assignments (i.e., false positives).

CornCyc4.0 and MaizeCyc2.2 are based on the same maize genome assembly version (B73 RefGen_v2), and reaction and pathway mapping were done using the Pathway Tools software suite that heavily uses an "encyclopedia" of pathways "from all domains of life" called MetaCyc [22]. CornCyc and MaizeCyc, however, were created by two different research groups based on their pipeline for enzymatic function assignments. In this work, we harnessed the availability of these two distinct metabolic pathway resources for maize in order to compare how initial enzymatic function assignments influence the final products that the biologists commonly use in their research.

Our results demonstrate that even though both CornCyc and MaizeCyc were constructed using the same gene model set and the same pathway assignment software, they have significantly different content. When we compared both databases in detail, we observed that MaizeCyc contains a larger number of annotated proteins whereas CornCyc covers a larger metabolic space having more compounds, reactions, and pathways.

We also extracted experimentally determined enzymatic function assignments from UniProt and analyzed how well these assignments were discovered by the computational pipelines used during the development of the resources. We defined performance measures such as precision and recall, and consolidated these results into a single F-score. F-score comparison demonstrates that though CornCyc coverage is more limited than that of MaizeCyc in terms of genes, its functional annotations are more stringent, and therefore more reliable for creating further hypotheses.

To conclude, computational pipelines used in the initial enzymatic function assignments and subsequent manual curation can have a large impact on the scope and range of the final metabolic pathway resources. The features of these pipelines and how they harness experimental data determine the final accuracy and quality of these resources. Finally, the accuracy of any metabolic pathway resource can be enhanced by dedicated and meticulous manual curation.

Author Contribution

TZS, JRW, and JAD conceived, designed, and coordinated the project. JRW developed and documented the software, and drafted the manuscript with TZS. JAD provided guidance on the software development and MLS on database curation of CornCyc and MaizeCyc. SYR and PZ created and curated CornCyc and provided their expertise in CornCyc and metabolic databases. All authors contributed to the manuscript writing process, and approved the final manuscript.

Acknowledgments

We thank the MaizeGDB team for sharing their insights and expertise. We thank Jacqueline D. Richter for her early contributions to the data extraction efforts. The material presented here is based upon work supported by the National Science Foundation grants EEC-0813570 and ABI-1062546 to JRW and JAD, IOS-1026003 to S.Y.R., Department of Energy grant DE-SC0008769 to S.Y.R., and National Institutes for Health grant U01 GM110699 to S.Y.R. This work was also supported by long-term funding from the U.S Department of Agriculture Agricultural Research Service to TZS and MLS. Any opinions, findings, and conclusions or recommendations expressed in the material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

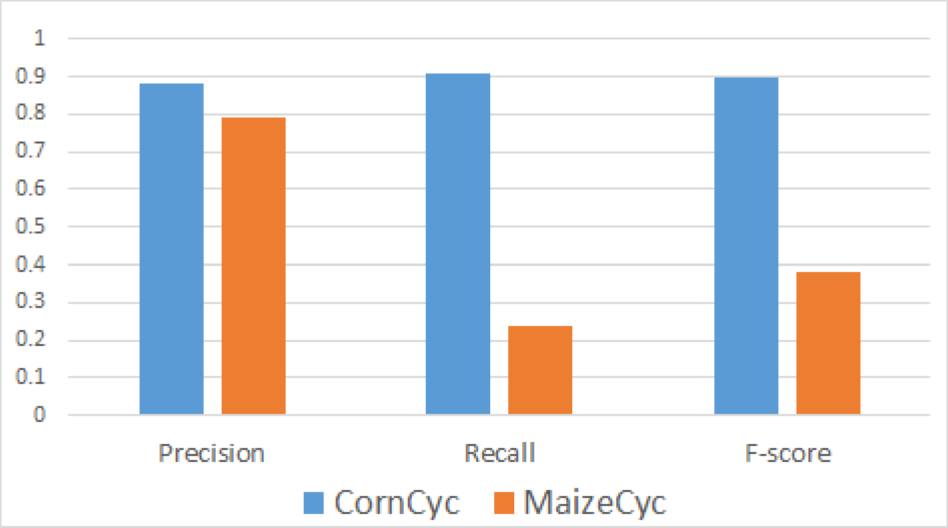


Figure 3.1 **Performance measure comparison between CornCyc 4.0 and MaizeCyc 2.2 based on 177 experimental protein functional annotations.**

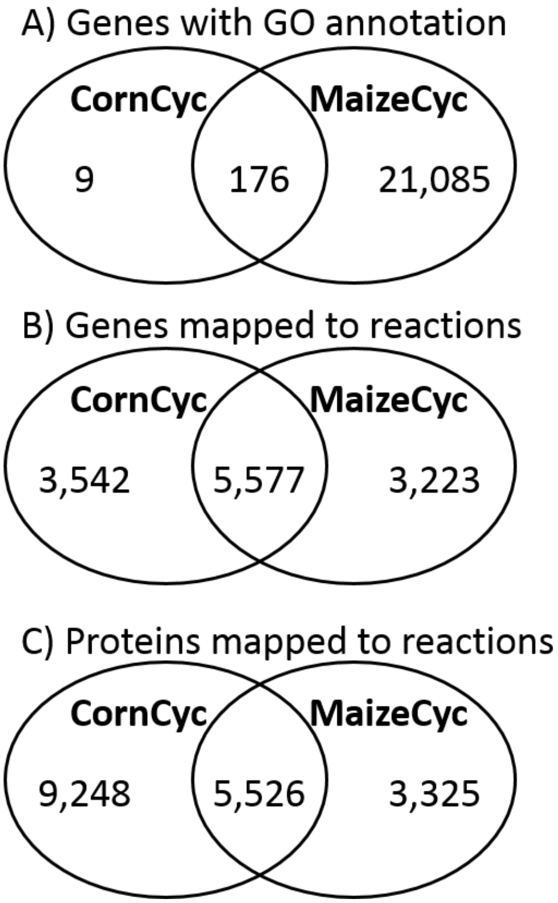


Figure 3.2 **Comparison of gene, protein, compound, reaction, and pathway statistics between CornCyc4.0 and MaizeCyc2.2.**

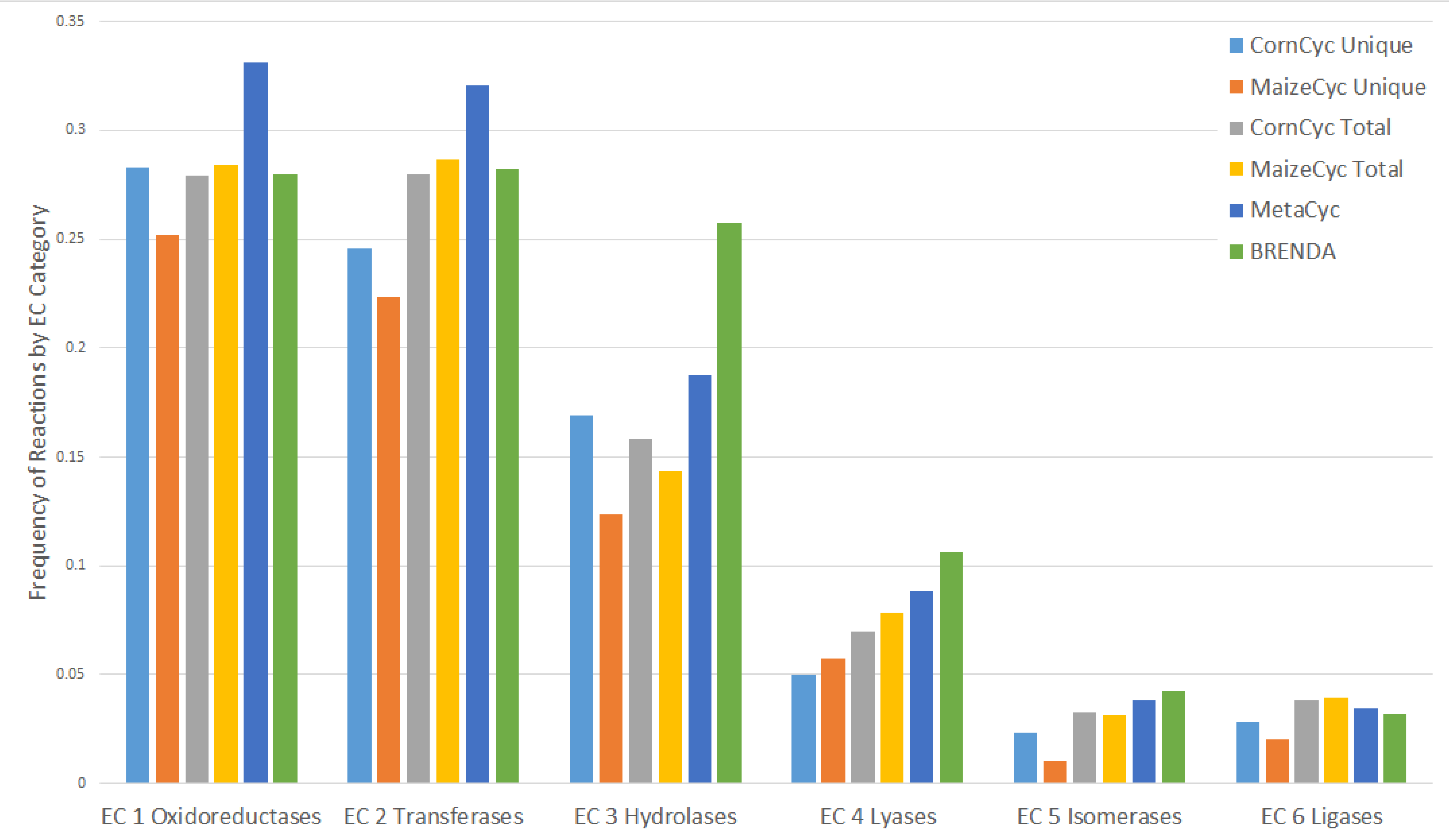


Figure 3.3 **Comparison of Reactions Sorted by EC Category between CornCyc 4.0, MaizeCyc 2.2, BRENDA (July 2015 Release), and MetaCyc 19.5.** For CornCyc and MaizeCyc, reactions with no EC category are not shown. CornCyc unique reactions refer to all reactions that were unique to the CornCyc when compared to MaizeCyc, and vice versa. For MetaCyc and BRENDA, all reactions, including those not found in plants, were included.

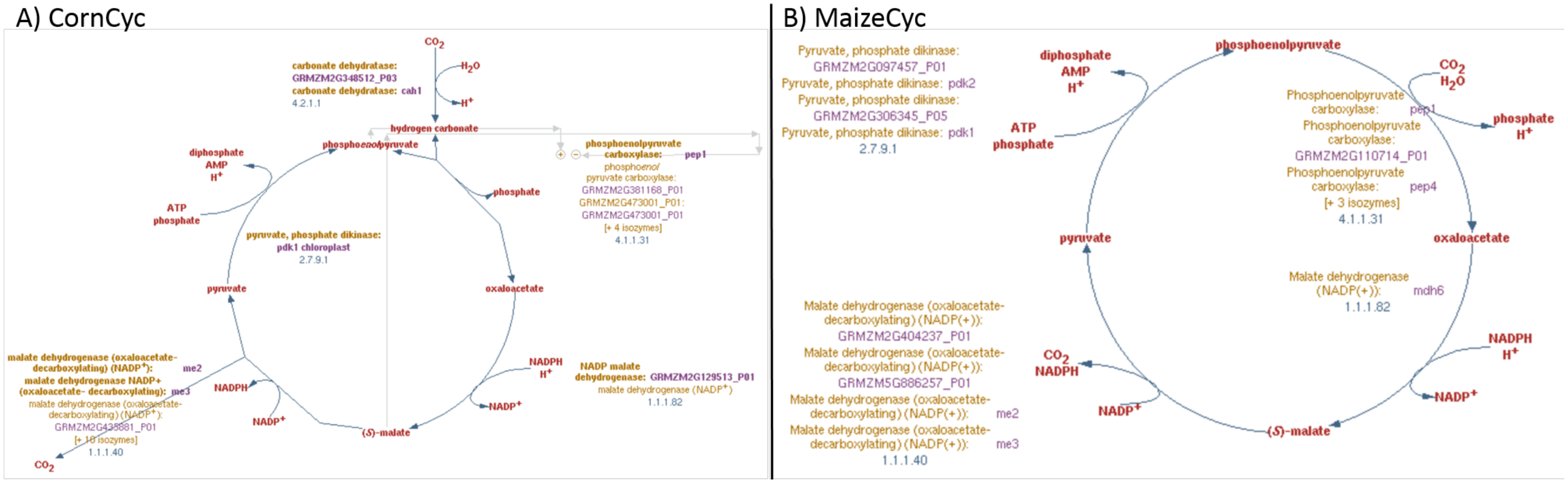

A) CornCyc
CO$_2$
carbonate dehydratase: GRMZM2G348512_P03
carbonate dehydratase: cah1
4.2.1.1
H$_2$O
H$^+$
hydrogen carbonate
phosphoenolpyruvate
diphosphate
AMP
H$^+$
ATP
phosphate
pyruvate, phosphate dikinase: pdk1 chloroplast
2.7.9.1
phosphate
phosphoenolpyruvate carboxylase: pep1
phosphoenol pyruvate carboxylase: GRMZM2G381168_P01 GRMZM2G473001_P01: GRMZM2G473001_P01
[+ 4 isozymes]
4.1.1.31
pyruvate
oxaloacetate
malate dehydrogenase (oxaloacetate-decarboxylating) (NADP$^+$): me2
malate dehydrogenase NADP+ (oxaloacetate- decarboxylating): me3
malate dehydrogenase (oxaloacetate-decarboxylating) (NADP$^+$): GRMZM2G435881_P01
[+ 10 isozymes]
1.1.1.40
NADPH
NADP$^+$
CO$_2$
(S)-malate
NADPH
H$^+$
NADP$^+$
NADP malate dehydrogenase: GRMZM2G129513_P01
malate dehydrogenase (NADP$^+$)
1.1.1.82
B) MaizeCyc
Pyruvate, phosphate dikinase: GRMZM2G097457_P01
Pyruvate, phosphate dikinase: pdk2
Pyruvate, phosphate dikinase: GRMZM2G306345_P05
Pyruvate, phosphate dikinase: pdk1
2.7.9.1
diphosphate
AMP
H$^+$
ATP
phosphate
phosphoenolpyruvate
CO$_2$
H$_2$O
phosphate
H$^+$
Phosphoenolpyruvate carboxylase: pep1
Phosphoenolpyruvate carboxylase: GRMZM2G110714_P01
Phosphoenolpyruvate carboxylase: pep4
[+ 3 isozymes]
4.1.1.31
pyruvate
oxaloacetate
Malate dehydrogenase (NADP(+)): mdh6
1.1.1.82
Malate dehydrogenase (oxaloacetate-decarboxylating) (NADP(+)): GRMZM2G404237_P01
Malate dehydrogenase (oxaloacetate-decarboxylating) (NADP(+)): GRMZM5G886257_P01
Malate dehydrogenase (oxaloacetate-decarboxylating) (NADP(+)): me2
Malate dehydrogenase (oxaloacetate-decarboxylating) (NADP(+)): me3
1.1.1.40
CO$_2$
NADPH
NADP$^+$
NADPH
H$^+$
NADP$^+$
(S)-malate

Figure 3.4 **Comparison of Pathways between CornCyc 4.0 and MaizeCyc 2.2, showing an example of pathway variation.** Shown are the C4 photosynthetic carbon assimilation pathways for CornCyc and MaizeCyc. Notwithstanding the differences in enzyme assignments, the core pathway supports the same function for both variants. The main difference is that CornCyc treats hydrogen carbonate formation as a separate step, while MaizeCyc groups it into the phosphoenolpyruvate to oxaloacetate step.

Table 3.1 Prediction performance of CornCyc 4.0 and MaizeCyc 2.2.

	True Positive	**False Positive**	**False Negative**
MaizeCyc	42	11	129
CornCyc	138	19	13

Table 3.2 Comparison of reaction and EC number statistics for all reactions in CornCyc 4.0 and MaizeCyc 2.2.

	CornCyc	**Overlap**	**MaizeCyc**
Reactions	1,275	1,686	591
Oxidoreductases (EC 1)	361	463	149
Transferases (EC 2)	313	514	132
Hydrolases (EC 3)	215	254	73
Lyases (EC 4)	64	142	34
Isomerases (EC 5)	30	65	6
Ligases (EC 6)	36	78	12
Unclassified (No EC Number)	256	121	185

Bibliography

[1] Consortium, T.G.O.: Gene Ontology Consortium: going forward. Nucleic Acids Research **43**(D1), 1049–1056 (2015). doi:10.1093/nar/gku1179

[2] Webb, E.C.: Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes. (Ed. 6), 863 (1992)

[3] Wittig, U., Beuckelaer, A.D.: Analysis and comparison of metabolic pathway databases. Briefings in Bioinformatics **2**(2), 126–142 (2001). doi:10.1093/bib/2.2.126

[4] Lee, T.J., Pouliot, Y., Wagner, V., Gupta, P., Stringer-Calvert, D.W., Tenenbaum, J.D., Karp, P.D.: BioWarehouse: a bioinformatics database warehouse toolkit. BMC Bioinformatics **7**, 170 (2006). doi:10.1186/1471-2105-7-170

[5] Kumar, A., Suthers, P.F., Maranas, C.D.: MetRxn: a knowledgebase of metabolites and reactions spanning metabolic models and databases. BMC Bioinformatics **13**(1), 6 (2012). doi:10.1186/1471-2105-13-6

[6] Altman, T., Travers, M., Kothari, A., Caspi, R., Karp, P.D.: A systematic comparison of the MetaCyc and KEGG pathway databases. BMC Bioinformatics **14**(1), 112 (2013). doi:10.1186/1471-2105-14-112

[7] Schnable, P.S., Ware, D., Fulton, R.S., Stein, J.C., Wei, F., Pasternak, S., Liang, C., Zhang, J., Fulton, L., Graves, T.A., Minx, P., Reily, A.D., Courtney, L., Kruchowski, S.S., Tomlinson, C., Strong, C., Delehaunty, K., Fronick, C., Courtney, B., Rock, S.M., Belter, E., Du, F., Kim, K., Abbott, R.M., Cotton, M., Levy, A., Marchetto, P., Ochoa, K., Jackson, S.M., Gillam, B., Chen, W., Yan, L., Higginbotham, J., Cardenas, M., Waligorski,

J., Applebaum, E., Phelps, L., Falcone, J., Kanchi, K., Thane, T., Scimone, A., Thane, N., Henke, J., Wang, T., Ruppert, J., Shah, N., Rotter, K., Hodges, J., Ingenthron, E., Cordes, M., Kohlberg, S., Sgro, J., Delgado, B., Mead, K., Chinwalla, A., Leonard, S., Crouse, K., Collura, K., Kudrna, D., Currie, J., He, R., Angelova, A., Rajasekar, S., Mueller, T., Lomeli, R., Scara, G., Ko, A., Delaney, K., Wissotski, M., Lopez, G., Campos, D., Braidotti, M., Ashley, E., Golser, W., Kim, H., Lee, S., Lin, J., Dujmic, Z., Kim, W., Talag, J., Zuccolo, A., Fan, C., Sebastian, A., Kramer, M., Spiegel, L., Nascimento, L., Zutavern, T., Miller, B., Ambroise, C., Muller, S., Spooner, W., Narechania, A., Ren, L., Wei, S., Kumari, S., Faga, B., Levy, M.J., McMahan, L., Buren, P.V., Vaughn, M.W., Ying, K., Yeh, C.-T., Emrich, S.J., Jia, Y., Kalyanaraman, A., Hsia, A.-P., Barbazuk, W.B., Baucom, R.S., Brutnell, T.P., Carpita, N.C., Chaparro, C., Chia, J.-M., Deragon, J.-M., Estill, J.C., Fu, Y., Jeddeloh, J.A., Han, Y., Lee, H., Li, P., Lisch, D.R., Liu, S., Liu, Z., Nagel, D.H., McCann, M.C., SanMiguel, P., Myers, A.M., Nettleton, D., Nguyen, J., Penning, B.W., Ponnala, L., Schneider, K.L., Schwartz, D.C., Sharma, A., Soderlund, C., Springer, N.M., Sun, Q., Wang, H., Waterman, M., Westerman, R., Wolfgruber, T.K., Yang, L., Yu, Y., Zhang, L., Zhou, S., Zhu, Q., Bennetzen, J.L., Dawe, R.K., Jiang, J., Jiang, N., Presting, G.G., Wessler, S.R., Aluru, S., Martienssen, R.A., Clifton, S.W., McCombie, W.R., Wing, R.A., Wilson, R.K.: The b73 maize genome: Complexity, diversity, and dynamics. Science **326**(5956), 1112–1115 (2009). PMID: 19965430

[8] Karp, P.D., Paley, S., Romero, P.: The Pathway Tools software. Bioinformatics **18**(suppl 1), 225–232 (2002)

[9] Chae, L., Lee, I., Shin, J., Rhee, S.Y.: Towards understanding how molecular networks evolve in plants. Current Opinion in Plant Biology **15**(2), 177–184 (2012)

[10] Chae, L., Kim, T., Nilo-Poyanco, R., Rhee, S.Y.: Genomic signatures of specialized metabolism in plants. Science (New York, N.Y.) **344**(6183), 510–513 (2014). doi:10.1126/science.1252076

[11] Zhang, P., Dreher, K., Karthikeyan, A., Chi, A., Pujar, A., Caspi, R., Karp, P., Kirkup, V., Latendresse, M., Lee, C., Mueller, L.A., Muller, R., Rhee, S.Y.: Creation of a Genome-Wide Metabolic Pathway Database for Populus trichocarpa Using a New Approach for Reconstruction and Curation of Metabolic Pathways for Plants. Plant Physiology **153**(4), 1479–1491 (2010). doi:10.1104/pp.110.157396

[12] Sen, T.Z., Andorf, C.M., Schaeffer, M.L., Harper, L.C., Sparks, M.E., Duvick, J., Brendel, V.P., Cannon, E., Campbell, D.A., Lawrence, C.J.: MaizeGDB becomes ’sequence-centric’. Database: The Journal of Biological Databases and Curation **2009** (2009)

[13] Andorf, C.M., Cannon, E.K., Portwood, J.L., Gardiner, J.M., Harper, L.C., Schaeffer, M.L., Braun, B.L., Campbell, D.A., Vinnakota, A.G., Sribalusu, V.V., Huerta, M., Cho, K.T., Wimalanathan, K., Richter, J.D., Mauch, E.D., Rao, B.S., Birkett, S.M., Sen, T.Z., Lawrence-Dill, C.J.: MaizeGDB update: new tools, data and interface for the maize model organism database. Nucleic Acids Research **44**(D1), 1195–1201 (2016). doi:10.1093/nar/gkv1007

[14] Monaco, M.K., Sen, T.Z., Dharmawardhana, P.D., Ren, L., Schaeffer, M., Naithani, S., Amarasinghe, V., Thomason, J., Harper, L., Gardiner, J., Cannon, E.K., Lawrence, C.J., Ware, D., Jaiswal, P.: Maize metabolic network construction and transcriptome analysis. The Plant Genome **6**(1), 0 (2013)

[15] Cunningham, F., Amode, M.R., Barrell, D., Beal, K., Billis, K., Brent, S., Carvalho-Silva, D., Clapham, P., Coates, G., Fitzgerald, S., Gil, L., Girn, C.G., Gordon, L., Hourlier, T., Hunt, S.E., Janacek, S.H., Johnson, N., Juettemann, T., Khri, A.K., Keenan, S., Martin, F.J., Maurel, T., McLaren, W., Murphy, D.N., Nag, R., Overduin, B., Parker, A., Patricio, M., Perry, E., Pignatelli, M., Riat, H.S., Sheppard, D., Taylor, K., Thormann, A., Vullo, A., Wilder, S.P., Zadissa, A., Aken, B.L., Birney, E., Harrow, J., Kinsella, R., Muffato, M., Ruffier, M., Searle, S.M.J., Spudich, G., Trevanion, S.J., Yates, A., Zerbino, D.R., Flicek, P.: Ensembl 2015. Nucleic Acids Research **43**(D1), 662–669 (2015). doi:10.1093/nar/gku1010

[16] Slater, G.S.C., Birney, E.: Automated generation of heuristics for biological sequence comparison. BMC Bioinformatics **6**, 31 (2005). doi:10.1186/1471-2105-6-31

[17] Van Hemert, J.L., Dickerson, J.A.: PathwayAccess: CellDesigner plugins for pathway databases. Bioinformatics **26**(18), 2345–2346 (2010)

[18] Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. Journal of Molecular Biology **215**(3), 403–410 (1990). doi:10.1016/S0022-2836(05)80360-2

[19] Yu, C., Zavaljevski, N., Desai, V., Reifman, J.: Genome-wide enzyme annotation with precision control: catalytic families (CatFam) databases. Proteins **74**(2), 449–460 (2009). doi:10.1002/prot.22167

[20] Claudel-Renard, C., Chevalet, C., Faraut, T., Kahn, D.: Enzyme-specific profiles for genome annotation: PRIAM. Nucleic Acids Research **31**(22), 6633–6639 (2003). doi:10.1093/nar/gkg847

[21] Consortium, T.U.: UniProt: a hub for protein information. Nucleic Acids Research **43**(D1), 204–212 (2015). doi:10.1093/nar/gku989

[22] Caspi, R., Altman, T., Billington, R., Dreher, K., Foerster, H., Fulcher, C.A., Holland, T.A., Keseler, I.M., Kothari, A., Kubo, A., Krummenacker, M., Latendresse, M., Mueller, L.A., Ong, Q., Paley, S., Subhraveti, P., Weaver, D.S., Weerasinghe, D., Zhang, P., Karp, P.D.: The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. Nucleic Acids Research **42**(D1), 459–471 (2014). doi:10.1093/nar/gkt1103

[23] Chang, A., Schomburg, I., Placzek, S., Jeske, L., Ulbrich, M., Xiao, M., Sensen, C.W., Schomburg, D.: BRENDA in 2015: exciting developments in its 25th year of existence. Nucleic Acids Research, 1068 (2014). doi:10.1093/nar/gku1068

[24] Schnable, J.C., Freeling, M.: Genes Identified by Visible Mutant Phenotypes Show Increased Bias toward One of Two Subgenomes of Maize. PLoS ONE **6**(3), 17855 (2011). doi:10.1371/journal.pone.0017855

[25] Frey, M., Schullehner, K., Dick, R., Fiesselmann, A., Gierl, A.: Benzoxazinoid biosynthesis, a model for evolution of secondary metabolic pathways in plants. Phytochemistry **70**(1516), 1645–1651 (2009). doi:10.1016/j.phytochem.2009.05.012

[26] Phillips, K.A., Skirpan, A.L., Liu, X., Christensen, A., Slewinski, T.L., Hudson, C., Barazesh, S., Cohen, J.D., Malcomber, S., McSteen, P.: vanishing tassel2 Encodes a Grass-Specific Tryptophan Aminotransferase Required for Vegetative and Reproductive Development in Maize[C][W][OA]. The Plant Cell **23**(2), 550–566 (2011). doi:10.1105/tpc.110.075267

[27] Walsh, J.R., Sen, T.Z., Dickerson, J.A.: A computational platform to maintain and migrate manual functional annotations for BioCyc databases. BMC Systems Biology **8**(1), 115 (2014). doi:10.1186/s12918-014-0115-1

4. MODELING THE EFFECT OF METABOLIC CONSTRAINTS ON TRANSCRIPTION FACTOR ACTIVITY LEVELS

A paper prepared for submission to a scholarly journal

Jesse R. Walsh and Julie A. Dickerson

Abstract

Computational models capable of predicting cellular phenotypes have become a standard method for several applications including genetic engineering, drug discovery, and pathway analysis. Constraint-based approaches can be used to determine the optimal route through a genome-scale metabolic network given defined cellular objectives and the stoichiometry of reactions catalyzed by the organism. As high-throughput methods continue to become more easily available, there is an increasing need for metabolic models which can integrate omics data to improve predictions of cellular function at the systems-level.

Current methods used to integrate regulatory information into constraint-based metabolic models cannot be used to investigate the regulatory states available to an organism. We describe a framework for integrating regulatory networks into existing metabolic models by extending the stoichiometric matrix to include transcriptional regulators to generate Regulation Enhanced Metabolic (REM) models. The models are capable of determining transcription factor activity ranges under a variety of metabolic conditions.

We demonstrate the feasibility of this approach by creating a REM model for *E. coli*. We first apply a series of regulatory constraints to an existing *E. coli* metabolic model, then analyze the regulatory states of the cell. We predict changes in transcription factor activities related to overproduction of short-chain fatty acid production. Finally, we describe a method

for determining the statistical significance of the predictions based on randomization of the regulatory network.

Introduction

As new applications for metabolic modeling continue to be developed, the scope and complexity of these models must grow to accommodate them [1]. In particular, there is a need for methods which integrate transcriptional regulatory and metabolic models. One recently proposed method demonstrates the utility of applying constraint-based methods to consider how regulated metabolic activity may exert some influence over the regulators themselves [2] by constraining regulatory states to those which facilitate a viable growth phenotype. Other recent studies [3, 4] have found regulatory interventions in *E. coli* which synergize with metabolic interventions predicted by constraint-based metabolic models. Such cases highlight the need for models which can integrate regulatory and metabolic networks to not only predict optimal metabolic states, but predict optimal regulatory states as well.

There are several existing methods, such as PROM [5] and E-Flux [6], which are able to predict cellular growth phenotypes with greater accuracy by including expression data and/or transcriptional regulation. These methods constrain the solution space of predicted metabolic fluxes by limiting flux through reactions based on a transcriptional regulatory state. These tools use expression data to define a relationship between gene expression and the reactions catalyzed by that gene's products such that they associate higher gene expression with greater metabolic flux. Such methods are not able to predict regulatory states as they require a single regulatory state to be defined when creating the model, and are thus unable to optimize the regulatory network in the context of a defined metabolic network.

Here we describe a novel approach to integrating the transcriptional regulatory and metabolic networks of *E. coli* to form a Regulation Enhanced Metabolic (REM) model. The method makes use of predictive regulatory network models to generate Linear Programming (LP) formulated constraints which are then linked to reactions in the metabolic network. The combined model optimizes both regulatory and metabolic states. We show the utility of this model by predicting regulatory states for E. coli during overproduction of short-chain fatty acid.

Methods

Overview of Model Integration

This method takes in a regulatory network and integrates it with a corresponding metabolic network (Figure 4.1). The relative influence of a regulator against a particular target gene is computed through analysis of gene expression data. Regulators are matched to reactions using the Gene Protein Reaction (GPR) rules provided in the metabolic model. An equation describing how transcriptional regulators influence the target reaction is generated for each reaction. These are appended to the stoichiometric matrix of the metabolic model as follows: 1) a column is created for each transcription factor, 2) a row representing the transcription factor constraints affecting a given reaction is added for each reaction in the model, and 3) the weights representing the regulatory response of a reaction to changes in transcription factor activity is added to the corresponding row/column of the matrix. This method of integration introduces a set of variables representing transcriptional regulator activities which can be optimized using existing methods for analyzing constraint-based networks.

Both reactions and transcription factor activities can be compared between a wild-type strain and a hypothetical mutant strain. For cases where a reaction flux is different in the mutant strain, existing metabolic engineering frameworks [7, 8] can predict interventions which will transform the wild-type strain into the mutant strain. Since regulatory constraints are not stoichiometrically accurate due to the stochastic nature of transcriptional regulation, we define a novel method to determine the significance of changes in transcriptional regulatory activities.

Regulatory Network Inference

The network inference methods used in this research, cMonkey and Inferelator [9, 10], have previously been used to infer the regulatory network of several organisms including H. salinarum [11] and Saccharomyces cerevisiae [12]. The cMonkey and Inferelator algorithms together form a pipeline which takes in a compendium of gene expression data and outputs a regulatory network model. The cMonkey algorithm performs biclustering as a way to address complexity of the data. Biclusters group putatively co-regulated genes (rows) across similar experimental conditions (columns) based on coherence in expression data and additional information including gene networks and motif analysis. Inferelator is a regression algorithm which performs additive linear regression of known regulators across the set of biclusters generated by cMonkey. Inferelator selects a parsimonious model by performing L1 shrinkage.

Network Integration

Metabolic models can simulate metabolic phenotypes under steady state conditions. Constraint-based methods have been available for decades [13], however we refer to [14] for detailed information on the Flux Balance Analysis (FBA) method and Flux Variability Analysis (FVA) method. Briefly, FBA is framed as a linear programming problem:

$$
\begin{aligned}
&max\ c^T v \\
&subject\ to \begin{cases} \quad Sv = 0 \\ a_j \leq v_j \leq b_j \end{cases}
\end{aligned}
\tag{4.1}
$$

where S is the stoichiometric matrix, v is the vector of reaction fluxes, a is the lower bound for reaction j, and b is the upper bound for reaction j. The objective can be either maximization or minimization of the objective function, which is typically defined using c^T to maximize the biomass equation. Other objective functions can be specified, such as MOMA [15] or the production of a specific compound. FVA simply calls the FBA method repeatedly to maximize, then minimize each reaction in the network. This gives a lower and upper flux value for each reaction.

Our proposed method modifies the FBA formulation by adding additional constraints tying flux through a reaction to the transcription factor activities of the genes associated with that reaction:

$$max\ c^T v$$
$$subject\ to \begin{cases} Sv = 0 \\ v_j < \beta_1 * TF_1 + \beta_2 * TF_2 + ...\beta_i * TFi \\ a_j \leq v_j \leq b_j \\ 0 \leq TF_i \leq 1000 \end{cases} \tag{4.2}$$

where the additional constraint links the flux of reaction j to the activation of its enzyme as defined by the combination of regulatory response parameters β and regulator activities TF. The variable TF_i represents the activity of transcription factor i. The variable β_i represents the regulator response strength of the transcription factor to this specific target gene. These values are determined in a data-driven manner using network inference tools such as Inferelator [9].

The regulator response variables predicted by Inferelator are given in the form:

$$\beta_1 * TF_1 + \beta_2 * TF_2 + ...\beta_i * TF_i = TargetExpression_j \tag{4.3}$$

We define a relationship between transcription factor activation of an enzyme's expression and flux through the enzyme's reaction, given by the equation:

$$\beta_1 * TF_1 + \beta_2 * TF_2 + ...\beta_i * TF_i = v_j \tag{4.4}$$

The TF values represent transcription factor activity for the transcription factor, while the β values represent the affinity for the promoters of the genes catalyzing the reaction. A negative β value represents gene repression.

The lower and upper bounds for regulators are set to 0 and 1000 respectively. The lower values represent the fact that a transcription factor cannot have a negative activity. The upper values are chosen to be arbitrarily large such that the new regulatory constraints can span the scope of the existing metabolic states.

Randomization Method for Identifying Significant Regulatory Changes

Using the REM model, transcriptional regulatory activities can be predicted using FBA and FVA using the same optimization techniques as are used for metabolic fluxes. FVA is used to determine both the upper and lower range of each transcription factor activity in the system. Changes in transcription factor activity ranges under mutant conditions imply a relationship between the regulator and the mutant phenotype. Several interesting outcomes are possible when comparing transcription factor activity ranges. If the regulator is predicted to turn on or off, the interaction is considered interesting. If the activity increases or decreases, the interaction may be interesting if the shift is great enough. While changes in metabolic flux are defined stoichiometrically and represent a difference defined in grams [dry weight] per hour, changes in transcription factor activities are unitless and cannot be compared directly. Therefore, we propose a randomization method for identifying significant transcriptional regulatory shifts in the REM model (Figure 4.2).

In order to determine when a regulatory shift is significant, we must determine if the network response to the selected regulatory model is significant. This method generates semi-randomized networks such that the regulatory response variables are randomized while the network structure is not. New regulatory response variables are sampled from the empirical distribution of regulatory response variables predicted by the regulatory network inference algorithm in order to generate randomized regulatory models. Positive values represent transcription factor activation of a gene, while negative values represent repression of a gene. A regulatory response of 0 represents no interaction between a transcription factor and a gene. This method does not allow regulatory interactions to be broken or created, only the strength or the direction of the interaction can change.

Each randomized regulatory model is integrated with the metabolic model to generate a series of semi-random REM models. FVA is used to calculate the feasible range for each transcription factor activity in order to determine each regulatory shift under the mutant condition. By comparing the predicted change to the those generated by the randomized models, we can compute how likely the predicted value is to have been found at random using a two-tailed probability calculation. Predicted values that are significantly greater than the random values can be interpreted to mean that the represented transcription factor needs to become more active, while predicted values significantly lower than the random values represent transcription factors that need to become less active.

Results

Case Study: Short Chain Fatty Acid Production in *E. coli*

We applied our method to generate a Regulation Enhanced Metabolic (REM) model for the overproduction of short-chain fatty acids in the organism *E. coli* MG1655. cMonkey found 430 biclusters across 4,266 genes and 466 conditions. Each bicluster contained between 7 and 34 genes (mean of 20 genes per cluster) and were associated with between 19 and 284 experimental conditions (mean of 233 conditions per cluster). Inferelator assigned no regulators to 21 of the biclusters, and generated 67,709 transcription factor to gene regulatory interactions across the remaining biclusters. There were an average of 7.9 predictors per bicluster.

These interactions represent the regulatory response of the target genes to the expression levels of their regulators. When Inferelator determined that a regulator could invoke different levels of response in the target gene under different conditions, two or more interactions between the same regulator and gene were created. For the purposes of this work, we only consider the first interaction, although averaging the interactions may be more representative of the interaction under novel conditions.

Regulator response values produced by Inferelator ranged from -0.3933184 to 0.9019466. This network is interpreted as a set of possible regulatory interactions that may include both direct and indirect regulation. Comparison with RegulonDB [16] gold standard interaction set showed a low recovery rate for true interactions. RegulonDB contains 4,269 transcription factor to gene interactions. The Inferelator recovered 11% (474) of these interactions, however the recovered interactions are on average stronger than then those not present in RegulonDB (p-value 2.2e-16). This suggests that removing very weak interactions may improve the quality of the network. A visual inspection of the distribution of values did not indicate a clear threshold to designate as a "weak" interaction, therefore we did not remove any weak interactions.

We compare the regulatory state for E. coli growth to the regulatory state for short-chain fatty acid (C6) overproduction. The base iAF1260 [17] model was modified to include fatty acid transport reactions prior to integration with the regulatory network. C6 production was simulated by maximizing the C6 export reaction "EX_hxa(e)". Flux Variability Analysis was used to compare the regulatory ranges available to the model for both wild-type growth and C6 export with no growth.

For each transcriptional regulator, a range was calculated representing the feasible activity levels of that regulator. By comparing the overproduction mutant to the wild-type strain, changes in regulatory state become apparent. Differences in ranges are expressed as a unitless value. These values represent transcriptional regulator activity on a positive scale from 0 to 1000. We found 14 regulators that shifted from off to on, and 12 that shifted from on to off (Table 4.1).

There are several genes whose activity is known to be related to fatty acid production. Previous studies [4] have found that deletion of fabR and up-regulation of fadR can improve fatty acid production. Interestingly, our model predicted the deletion of fadR and up-regulation of fabR. The regulators gadW and gadX positively influence the acid resistance system in *E. coli* are also thought to improve tolerance to high fatty acid production [18]. Our model agrees with the previous study, predicting both of these transcription factors to be up-regulated. In order to determine if the regulatory changes predicted by this model were significant, we applied our randomization method to generate 124 simulations comparing C6 overproduction to wild-type growth. For each simulation, we collected the upper and lower range for each transcription factor. Predictions for all transcription factors are available in Appendix Table C.1.

Discussion

Impact of Network Inference Method on Model Quality

The selection of the Inferelator network inference algorithm for use in generating REM models was motivated by several factors, including maturity of the algorithm and the fact that Inferelator performs a simple linear regression allowing for a more direct integration with the metabolic model. However, it is difficult to characterize the impact this choice had on the quality and accuracy of the resulting model. Since Inferelator infers both direct and indirect regulatory influences, it includes many interactions which do not represent direct binding of the transcription factor to the target gene (or gene operon). In fact, Inferelator recovered very few known direct interactions in *E. coli*. It is possible that the use of a network inference algorithm which only considers known direct regulatory interactions may improve the ability for our REM model to predict direct regulatory responses such as those for fabR and fadR in C6 overproduction.

The results produced during the C6 overproduction case study were encouraging but leave room for improvement. Most notable was the fact that fadR and fabR regulators were predicted to have the opposite effect of what has been shown in biological studies. These regulators directly regulate the fatty acid pathways, which makes them obvious targets of interest in a study of fatty acid production. While our model did not correctly predict the direct fatty acid pathway regulators, it was able to correctly identify the indirect effects of the gad acid resistance system by predicting up-regulation of gadW and gadX.

Network Randomization

The regulatory network model represents both the interaction between a transcription factor and its target as well as the regulatory response of the activation or repression. Biologically, the regulatory response variables attempt to capture the interaction between various regulators on a single target gene. A regulatory response may vary under different environmental and cellular conditions, however these dynamics are not truly represented in a linearly regressed model. Randomization of the regulatory network in this proposed method randomizes the regulatory response variables in order to test the sensitivity of the REM model to changes in regulatory responses.

It may be interesting to also randomize the structure of the regulatory network. Biologically, randomizing the regulatory network would test the model's sensitivity to the network structure predicted by the regulatory network inference algorithm. Our efforts to generate randomization in the network structure were only partially successful. We found that changes in the regulatory network structure frequently produced infeasible models, which increased the difficulty of producing a significant number of samples for testing significance of regulatory shifts as well as introduced additional challenges interpreting such randomization. It may be interesting in future work to introduce "soft" regulatory constraints that can be relaxed in order to improve model feasibility during the randomization testing.

Conclusion

One of the benefits of an integrated constraint-based regulatory and metabolic network model has over simple network analysis is that system level constraints may elucidate non-intuitive interactions important for a metabolic function of interest. For example, when over-producing fatty acids, it is fairly intuitive that the fatty acid biosynthesis and degradation regulators may have a strong influence over the flow of material to fatty acid production. The direct interaction between these regulators and target reactions of interest make them obvious targets of interest. However, producing fatty acids may lead to additional constraints on other parts of the network, such as the acid resistance system. Such interactions might be considered non-intuitive as there is not a direct link between the fatty acid reactions and these regulators. Such interactions are only visible when taken in context of the whole system.

In this study we present a novel way to predict both regulatory and metabolic states by integrating both networks into a constraint-based model. We additionally present a method for determining the significance of predictions made with this model. We demonstrated the feasibility of our method by predicting the regulatory states of *E. coli* during fatty acid over-production and by applying our randomization method to this model in order to determine significance of the predicted regulatory shifts.

Materials

Mapping Inferred Regulatory Network to iAF1260 Reactions

Inferelator generates a predictive regulatory network based on the biclusters produced by cMonkey. This network links transcription factors to target biclusters. In order to integrate the regulation into the metabolic model, we require a network which links transcription factors to metabolic reactions. We first expand the network such that every gene in the bicluster is regulated by every transcription factor linked to that bicluster. Essentially, this duplicates each column (bicluster) once for each gene in the bicluster, resulting in a network of transcription factors linked to genes.

Frequently, a single transcription factor will be linked to multiple clusters containing the same gene. We only consider the first link (in numerical order by bicluster). The transcription factor to gene links are then mapped to iAF1260 reactions using Gene-Protein-Reaction (GPR) rules. GPR rules describe the relationship between genes and reactions and are used to indicate when multiple genes are associated with a reaction (i.e. protein complexes, isozymes, etc.). We consider only the first gene mentioned in each GPR rule in order to simplify the mapping process.

Identification of Transcription Factors

The complete list of transcription factors in *E. coli* MG1655 was obtained from EcoCyc [19]. This list was used as input to the cMonkey algorithm. This list also served as a map to convert between gene "B-number" IDs and gene common names.

Gene Expression Data

We use the Many Microbe Microarray Database M3D [20] (version 4, build 6) normalized expression data for *E. coli* publically available at `http://m3d.mssm.edu/`. This dataset provides a large array of experimental conditions while mitigating variations in platform and chipsets.

Metabolic Model of *E. coli*

While the newer model iJO1366 [21] is available, the authors report both an increase in scope and a decrease in prediction accuracy. The iAF1260 model [17] of E. coli metabolism was used both for the higher accuracy and to provide a consistent comparison to previous work. This model was loaded into the Matlab environment using COBRA Toolbox [22]. All subsequent simulations were performed using modified COBRA routines in Matlab with the Gurobi linear programming (LP) solver.

Definition of Difference Between Transcription Factor Activity Ranges

Flux Variability Analysis generates a range of feasible values by indicating an upper and a lower bound. All values between those bounds are considered feasible. There are several ways to indicate a difference between to ranges of values, however we settled on a calculation based on the splitting the difference the upper and lower bounds and comparing that value. We calculate a single value representing the shift in regulatory range using the formula:

$$regulatory\ shift = \frac{wt_{upperBound} - wt_{lowerBound}}{2} - \frac{mut_{upperBound} - mut_{lowerBound}}{2} \tag{4.5}$$

which calculates the difference between the center of the wild type regulatory range and the center of the mutant regulatory range for each regulator.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

JRW and JAD conceived, designed, and coordinated the project. JRW developed and documented the software, and drafted the manuscript. JAD provided advice and guidance on the software development and drafting of the manuscript. All authors read and approved the final manuscript.

Acknowledgements

The material presented here is based upon work supported by the National Science Foundation under Award No. EEC-0813570. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

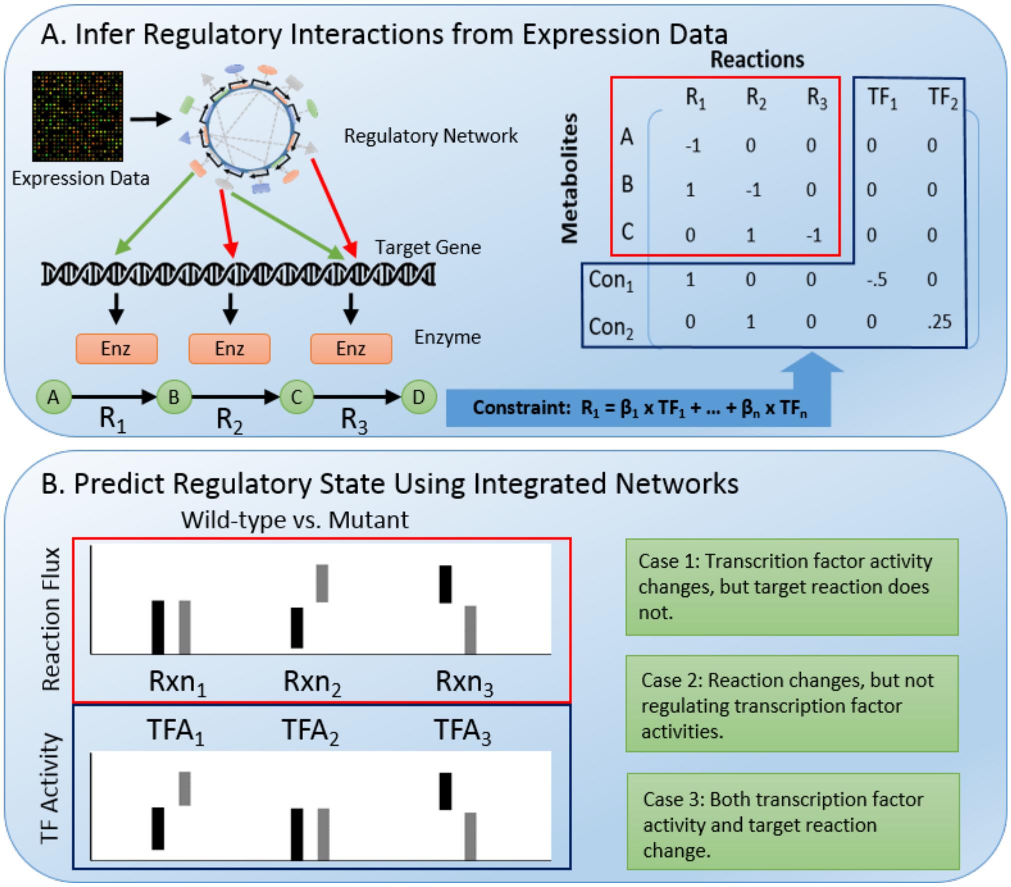

A. Infer Regulatory Interactions from Expression Data
Expression Data
Regulatory Network
Target Gene
Enzyme
Enz
Enz
Enz
A
B
C
D
R_1
R_2
R_3
Reactions
Metabolites
R_1 R_2 R_3 TF_1 TF_2
A -1 0 0 0 0
B 1 -1 0 0 0
C 0 1 -1 0 0
Con_1 1 0 0 -.5 0
Con_2 0 1 0 0 .25
Constraint: $R_1 = \beta_1 \times TF_1 + \ldots + \beta_n \times TF_n$
B. Predict Regulatory State Using Integrated Networks
Wild-type vs. Mutant
Reaction Flux
Rxn_1
Rxn_2
Rxn_3
TF Activity
TFA_1
TFA_2
TFA_3
Case 1: Transcrition factor activity changes, but target reaction does not.
Case 2: Reaction changes, but not regulating transcription factor activities.
Case 3: Both transcription factor activity and target reaction change.

Figure 4.1 Overview of Integration Method. A) A large compendium of expression data representing a broad array of experimental conditions is used to compute the regulatory network. Transcriptional regulators are related to metabolic reactions using the products of their target genes. We generate regulatory constraints from these interactions and integrate them into the stoichiometric matrix of the metabolic model. B) The metabolic and regulatory ranges are determined using Flux Variability Analysis (FVA). Comparison between the wild-type and mutant strain allows identification of reactions and regulator activities which must change in the mutant metabolic network. Several methods exist to predict interventions in metabolic reactions (case 2 and 3). In this paper, we focus on methods to predict interventions for transcriptional regulators (case 1 and 3).

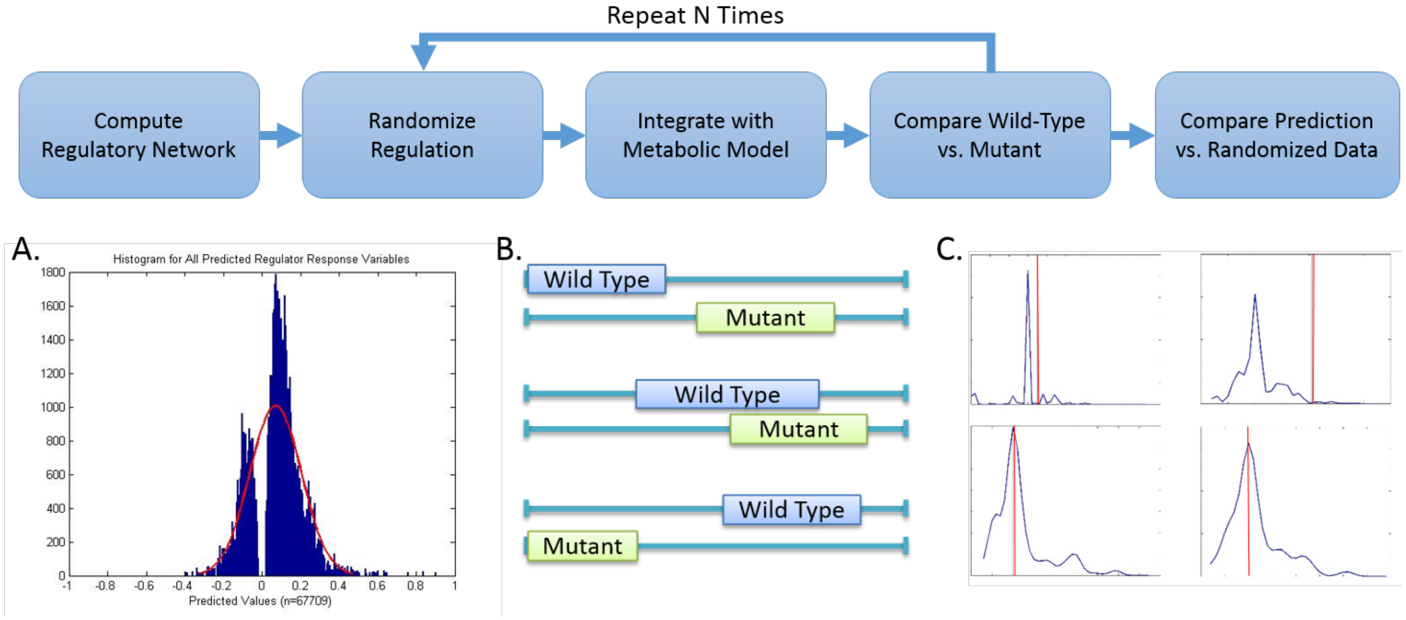

Repeat N Times
Compute Regulatory Network
Randomize Regulation
Integrate with Metabolic Model
Compare Wild-Type vs. Mutant
Compare Prediction vs. Randomized Data
A.
Histogram for All Predicted Regulator Response Variables
1800
1600
1400
1200
1000
800
600
400
200
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Predicted Values (n=67709)
B.
Wild Type
Mutant
Wild Type
Mutant
Wild Type
Mutant
C.

Figure 4.2 Overview of method for detecting significance of detected changes in transcription factor activity. A) An empirical distribution is computed for the regulatory response variables in the inferred regulatory network. These values may be positive (activator) or negative (repressor). A value of 0 indicates no relationship between a transcription factor and a target gene. For each iteration of the bootstrapping, the network weights are randomly replaced using values sampled from the original distribution of weights. Weights are allowed to change sign during this sampling, but not be broken. B) The integrated model is used to compare a wild-type condition to a mutant condition. The change in transcriptional regulatory activity is computed for each transcription factor over each iteraction of the procedure. C) The predicted change in transcription factor activity is compared to those from the randomized models in order to determine significance.

Table 4.1 Transcriptional regulators which were predicted to be turned on or turned off in the C6 overproduction *E. coli* mutant strain when compared with the wild-type growth condition.

Regulator Name	WT-Lower	WT-Upper	Mutant-Lower	Mutant-Upper	Change
bglJ	8.359491	8.618312	0	0.000345873	Turned off
csgD	2.10038	2.188148	0	0.000158402	Turned off
envR	5.551524	5.771021	0	0.000440512	Turned off
hupB	12.07819	12.13302	0	0.000240964	Turned off
iscR	6.252992	6.383168	0	0.000194162	Turned off
melR	16.03569	16.50607	0	0.000156418	Turned off
mhpR	7.950241	8.361945	0	0	Turned off
modE	1.138875	1.223445	0	0.000540972	Turned off
oxyR	9.925755	9.963603	0	0.000290612	Turned off
phoB	9.177673	9.332022	0	0.000157026	Turned off
phoR	3.318422	3.369362	0	0.397491085	Turned off
puuR	4.909688	5.114984	0	0.000949356	Turned off
cra	0	0.013704	10.37113	10.37535547	Turned on
crp	0	0.073513	4.067635	4.067749219	Turned on
dcuR	0	0.070202	7.131594	7.159531347	Turned on
fabR	0	0.02895	7.863744	7.864754432	Turned on
gadW	0	0.007195	32.48569	32.49279498	Turned on
gadX	0	0.00183	29.90562	29.91128164	Turned on
hyfR	0	0.003844	3.196545	3.196896824	Turned on
lsrR	0	0.049588	8.809217	8.814015115	Turned on
malI	0	0.014342	2.46917	2.477270986	Turned on
metR	0	0.071923	26.83228	26.83865725	Turned on
nanR	0	0.010803	6.571156	6.588342704	Turned on
rcnR	0	0.012832	8.624754	8.637378081	Turned on
sdiA	0	0.025528	10.5212	10.52197642	Turned on
ulaR	0	0.029808	16.55478	16.56569136	Turned on

Bibliography

[1] Bordbar, A., Monk, J.M., King, Z.A., Palsson, B.O.: Constraint-based models predict metabolic and associated cellular functions. Nature Reviews Genetics **15**(2), 107–120 (2014). doi:10.1038/nrg3643

[2] Chandrasekaran, S., Price, N.D.: Metabolic Constraint-Based Refinement of Transcriptional Regulatory Networks. PLoS Comput Biol **9**(12), 1003370 (2013). doi:10.1371/journal.pcbi.1003370

[3] Ranganathan, S., Tee, T.W., Chowdhury, A., Zomorrodi, A.R., Yoon, J.M., Fu, Y., Shanks, J.V., Maranas, C.D.: An integrated computational and experimental study for overproducing fatty acids in Escherichia coli. Metabolic Engineering **14**(6), 687–704 (2012). doi:10.1016/j.ymben.2012.08.008

[4] Tee, T.W., Chowdhury, A., Maranas, C.D., Shanks, J.V.: Systems metabolic engineering design: Fatty acid production as an emerging case study. Biotechnology and Bioengineering **111**(5), 849–857 (2014). doi:10.1002/bit.25205

[5] Chandrasekaran, S., Price, N.D.: Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in Escherichia coli and Mycobacterium tuberculosis. Proceedings of the National Academy of Sciences **107**(41), 17845–17850 (2010). doi:10.1073/pnas.1005139107

[6] Colijn, C., Brandes, A., Zucker, J., Lun, D.S., Weiner, B., Farhat, M.R., Cheng, T.-Y., Moody, D.B., Murray, M., Galagan, J.E.: Interpreting Expression Data with Metabolic Flux Models: Predicting Mycobacterium tuberculosis Mycolic Acid Production. PLoS Comput Biol **5**(8), 1000489 (2009). doi:10.1371/journal.pcbi.1000489

[7] Ranganathan, S., Suthers, P.F., Maranas, C.D.: OptForce: An Optimization Procedure for Identifying All Genetic Manipulations Leading to Targeted Overproductions. PLoS Comput Biol **6**(4), 1000744 (2010). doi:10.1371/journal.pcbi.1000744

[8] Burgard, A.P., Pharkya, P., Maranas, C.D.: Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. Biotechnology and Bioengineering **84**(6), 647–657 (2003). doi:10.1002/bit.10803

[9] Bonneau, R., Reiss, D.J., Shannon, P., Facciotti, M., Hood, L., Baliga, N.S., Thorsson, V.: The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. Genome Biology **7**(5), 36 (2006). doi:10.1186/gb-2006-7-5-r36

[10] Reiss, D.J., Baliga, N.S., Bonneau, R.: Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. BMC Bioinformatics **7**(1), 280 (2006). doi:10.1186/1471-2105-7-280

[11] Bonneau, R., Facciotti, M.T., Reiss, D.J., Schmid, A.K., Pan, M., Kaur, A., Thorsson, V., Shannon, P., Johnson, M.H., Bare, J.C., Longabaugh, W., Vuthoori, M., Whitehead, K., Madar, A., Suzuki, L., Mori, T., Chang, D.-E., DiRuggiero, J., Johnson, C.H., Hood, L., Baliga, N.S.: A Predictive Model for Transcriptional Control of Physiology in a Free Living Cell. Cell **131**(7), 1354–1365 (2007). doi:10.1016/j.cell.2007.10.053

[12] Danziger, S.A., Ratushny, A.V., Smith, J.J., Saleem, R.A., Wan, Y., Arens, C.E., Armstrong, A.M., Sitko, K., Chen, W.-M., Chiang, J.-H., Reiss, D.J., Baliga, N.S., Aitchison, J.D.: Molecular mechanisms of system responses to novel stimuli are predictable from public data. Nucleic Acids Research **42**(3), 1442–1460 (2014). doi:10.1093/nar/gkt938

[13] Varma, A., Palsson, B.O.: Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type Escherichia coli W3110. Applied and Environmental Microbiology **60**(10), 3724–3731 (1994)

[14] Orth, J.D., Thiele, I., Palsson, B..: What is flux balance analysis? Nature Biotechnology **28**(3), 245–248 (2010). doi:10.1038/nbt.1614

[15] Segr, D., Vitkup, D., Church, G.M.: Analysis of optimality in natural and perturbed metabolic networks. Proceedings of the National Academy of Sciences of the United States of America **99**(23), 15112–15117 (2002). doi:10.1073/pnas.232349399

[16] Salgado, H., Peralta-Gil, M., Gama-Castro, S., Santos-Zavaleta, A., Muiz-Rascado, L., Garca-Sotelo, J.S., Weiss, V., Solano-Lira, H., Martnez-Flores, I., Medina-Rivera, A., Salgado-Osorio, G., Alquicira-Hernndez, S., Alquicira-Hernndez, K., Lpez-Fuentes, A., Porrn-Sotelo, L., Huerta, A.M., Bonavides-Martnez, C., Balderas-Martnez, Y.I., Pannier, L., Olvera, M., Labastida, A., Jimnez-Jacinto, V., Vega-Alvarado, L., Del Moral-Chvez, V., Hernndez-Alvarez, A., Morett, E., Collado-Vides, J.: RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. Nucleic acids research **41**(Database issue), 203–213 (2013). PMID: 23203884 PMCID: PMC3531196

[17] Feist, A.M., Henry, C.S., Reed, J.L., Krummenacker, M., Joyce, A.R., Karp, P.D., Broadbelt, L.J., Hatzimanikatis, V., Palsson, B..: A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Molecular Systems Biology **3** (2007). doi:10.1038/msb4100155

[18] Royce, L.A., Boggess, E., Fu, Y., Liu, P., Shanks, J.V., Dickerson, J., Jarboe, L.R.: Transcriptomic Analysis of Carboxylic Acid Challenge in Escherichia coli: Beyond Membrane Damage. PLoS ONE **9**(2), 89580 (2014). doi:10.1371/journal.pone.0089580

[19] Keseler, I.M., Collado-Vides, J., Santos-Zavaleta, A., Peralta-Gil, M., Gama-Castro, S., Muniz-Rascado, L., Bonavides-Martinez, C., Paley, S., Krummenacker, M., Altman, T., Kaipa, P., Spaulding, A., Pacheco, J., Latendresse, M., Fulcher, C., Sarker, M., Shearer, A.G., Mackie, A., Paulsen, I., Gunsalus, R.P., Karp, P.D.: EcoCyc: a comprehensive database of Escherichia coli biology. Nucleic Acids Research **39**(Database), 583–590 (2010). doi:10.1093/nar/gkq1143

[20] Faith, J.J., Driscoll, M.E., Fusaro, V.A., Cosgrove, E.J., Hayete, B., Juhn, F.S., Schneider, S.J., Gardner, T.S.: Many Microbe Microarrays Database: uniformly normalized

Affymetrix compendia with structured experimental metadata. Nucleic Acids Research **36**(Database issue), 866–870 (2008). doi:10.1093/nar/gkm815

[21] Orth, J.D., Conrad, T.M., Na, J., Lerman, J.A., Nam, H., Feist, A.M., Palsson, B.O.: A comprehensive genome-scale reconstruction of Escherichia coli metabolism[mdash]2011. Mol Syst Biol **7** (2011). doi:10.1038/msb.2011.65

[22] Becker, S.A., Feist, A.M., Mo, M.L., Hannum, G., Palsson, B.O., Herrgard, M.J.: Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox. Nat. Protocols **2**(3), 727–738 (2007). doi:10.1038/nprot.2007.99

5. GENERAL CONCLUSIONS

General Discussion

As advances to computational techniques and high-throughput technologies continue to accelerate the reliance of biological research on information and omics data, questions regarding the quality and value of our biological data will become increasingly critical to our understanding of biological processes. Our ability to integrate various types of data, both in terms of the biological processes they represent and the inherent levels of quality we assign this data, will have a significant impact on the future course of the field of bioinformatics and computational biology. In this dissertation, we have identified and contributed to three issues regarding the integration of biological data.

In Chapter 2, we proposed a method for merging a high-value subset of data into a heterogeneous database. We presented the software CycTools, which is able to externally modify a Pathway Genome Database (PGDB) through the Pathway Tools application programming interface (API). We demonstrated the utility of this tool by effecting the transfer of high-quality gene ontology (GO) annotations from one PGDB to another. We tested this process by migrating 147 manually curated GO annotations from MaizeCyc to CornCyc in order to improve the overall quality of CornCyc annotations.

The CycTools software was also designed to modify an existing PGDB by adding, removing, or changing content in order to align it to the known metabolic processes of a substrain of the organism represented by the existing PGDB. This feature is especially salient in metabolic engineering applications, as iterative modifications to an organisms metabolism are rapidly made and the performance of the mutant strain assessed. Directed mutations are usually in the form of a gene addition or deletion, resulting in a mutant strain which is highly related to the parent strain. By rapidly generating PGDB databases with these interventions, users benefit from the built-in analysis features provided by the PGDB, including but not limited to features such as omics overlays and metabolic modeling when analyzing the mutant strain.

The second area of data integration we addressed was the need for methods of explicitly handling high-value subsets of data present in the context of larger knowledge bases. In Chapter 3, we described computational approaches to evaluating the impact on the quality of biological pathway databases derived from computational enzymatic functional annotation pipelines. We analyzed existing pathway databases CornCyc [1] and MaizeCyc [2] in order to assess their genomic, metabolic, and pathway content. We reasoned that two databases built from the same gene model set and the same pathway inference software should attribute their differences to their unique enzymatic function annotation pipelines. Using software we developed, we were able to automatically map biological objects between the databases in order to quantify their similarities and differences, and assess their coverage of metabolic space through the use of Enzyme Commission annotations.

Our analysis showed that both corn databases were led to very different conclusions as a result of the differences in the methods used during their generation. However, while the similarities between CornCyc and MaizeCyc allowed for a direct comparison of their content, there were additional factors contributing to their differences besides their choice of enzymatic function prediction pipelines. Each database includes separate manually curated data. They were also derived from different versions of the MetaCyc database. A more powerful argument could be made for the effect of the computational pipeline by controlling these additional factors. However, by analyzing the publicly available versions of these databases, we were able to provide a stronger case for which database would better suit the needs of maize researchers.

The final area of data integration which we addressed was the need for computational models allowing for integrated analysis of multiple levels of omics data. Such models are needed in order to predict cellular functions under novel conditions. In Chapter 4, we proposed a novel computational model which integrates transcriptional regulatory and metabolic networks. This model provides researchers a unique perspective when investigating the interactions between systems-wide metabolic and regulatory networks by predicting the cell-wide transcriptional regulator activities consistent with a metabolic flux state. we demonstrated the feasibility of this method by applying it to the generation of short-chain fatty acids in *E. coli*. This model was in particular able to capture the interaction between fatty acid over-production and over-expression of the acid resistance system.

In order to determine when regulatory shifts predicted by the model are significant, we proposed a novel method of applying randomization to the integrated model in order to generate a randomized sample of regulatory shift data. This is a difficult problem as methods used to solve constraint-based metabolic models typically provide a single optimal solution and are not subject to randomly selected parameters. By randomizing the regulatory network prior to integrating it with the metabolic network, we were able to produce a series of semi-random networks which allowed for the sampling of possible regulatory shifts for each transcription factor. Significant shifts can then be determined by comparing the predicted regulatory shift to those in the randomized models.

Recommendations for Future Work

An accurate characterization of the difference in quality between two biological databases requires an accurate method of mapping similar biological entities between databases. While the software presented in Chapter 3 is able to make large-scale automated comparison between two similar databases, the process required specific knowledge of the database structures and idiosyncrasies. Such comparisons continue to largely rely on matching the names of objects, which can be complicated by seemingly innocent typographical changes such as the use of "alpha" vs. "α" or the inclusion of html-style markup tags. Especially troublesome are examples where "corrections" to a standardized identifier (such as a chemical InChI string, or the splitting of a reaction into multiple steps) is not propagated to databases using these identifiers.

The problem of updates and corrections to newer versions of MetaCyc also played an important role in the comparison between CornCyc and MetaCyc. It is likely that the fact both databases were generated using different versions of MetaCyc had an impact on the comparison between them. A more robust comparison of the computational enzymatic function prediction pipeline effects would involve recreating each database using the same MetaCyc version and without any manual curation. A study of this nature could then be extended by varying each step of the process, such as the initial gene model assembly or the reaction and pathway inference step.

In Chapter 4, understanding the complex interactions between gene expression and metabolic flux is crucial when designing regulatory constraints. We make the assumption of a direct, linear relationship such that an increase in gene expression always leads to an increase in the metabolic flux of the reactions catalyzed by that gene's products. This is almost certainly not the case, but remains a useful assumption. It allows the integrated model to be formulated as an Linear Programming (LP) problem. LP formulations are computationally tractable at the genome-scale for *E. coli*, and this assumption only requires gene expression data which is readily available.

It would be interesting to vary the regulatory network inference algorithm used in Chapter 4 to determine if a network of only known direct interactions could improve results when using the model. The cMonkey/Inferelator method includes many indirect interactions and performed poorly at recovering known direct interactions. This may have contributed to the model failing to correctly predict the direct transcriptional regulators fabR and fadR.

Bibliography

[1] Chae, L., Lee, I., Shin, J., Rhee, S.Y.: Towards understanding how molecular networks evolve in plants. Current Opinion in Plant Biology **15**(2), 177–184 (2012)

[2] Monaco, M.K., Sen, T.Z., Dharmawardhana, P.D., Ren, L., Schaeffer, M., Naithani, S., Amarasinghe, V., Thomason, J., Harper, L., Gardiner, J., Cannon, E.K., Lawrence, C.J., Ware, D., Jaiswal, P.: Maize metabolic network construction and transcriptome analysis. The Plant Genome **6**(1), 0 (2013)

APPENDIX A. CYCTOOLS USER GUIDE

Introduction

CycTools is an interface for accessing and updating data in a BioCyc family Pathway/Genome Database (PGDB). Access to the database provided by the Pathway Tools programmatic API. The JavaCycO library allows conversion of Java based commands into queries which can be understood by the Pathway Tools API. In order to use CycTools, a server running Pathway Tools and JavaCycO must be available. Because of the requirements of JavaCycO, this server must be running on a Unix-like environment, such as Ubuntu. JavaCycO requires the Java Runtime Environment (JRE) version 1.7 or higher on both the host and client machines. Because the server portion of this tool must be run on a linux-like OS, the installation guide is provided for Linux (Ubuntu) only. The client portion of this tool can be run on any OS that supports java 1.7 or above.

Installation Guide

While CycTools does not need to be installed to run, it depends on other software which must be installed before CycTools can be used. The general steps to using CycTools include:

1. Install Java SDK on Server

2. Install Pathway Tools on Server

3. Install JavaCycO on Server

After installation is complete, running CycTools will involve launching Pathway Tools in API mode, launching JavaCycO, then launching CycTools and connecting to the JavaCycO instance. The following instructions assume the machine is running Ubuntu 12.04.

Install Java JDK

Java 1.7 JDK (java development kit) or higher must be installed on both the host and client machine. For windows or Mac machines, follow the instructions here http://www.java.com/en/download/manual.jsp.

1. Check java version:

   - java -version

   Verify installation of version 1.7 or above of the JDK. The JRE will not have the tools necessary to compile JavaCycO.

2. Install java jdk

   - sudo apt-get install openjdk-7-jdk

3. Verify that the correct version of java is set as default

   - sudo update-alternatives –config java

   If only one java version is installed, no action is needed. If multiple versions are installed, select the java-7 jdk option (see Figure A.1).

Installing Pathway Tools

A local installation of Pathway Tools on a Unix-like system is required for accessing PGDB information through CycTools. The server that Pathway Tools is installed to can be the same machine that CycTools will be run from, or another machine accessible to the client. Installation instructions for Pathway Tools can be found on the pathway tools website. Note that while Pathway Tools is currently free for academic research purposes, a license request must be made to access the installer. At of the time of this writing, the request can be made here: http://biocyc.org/download-bundle.shtml. This guide assumes Pathway Tools version 17.5 is installed using the default settings and file locations as a single user (i.e. not as the Administrator).

Installing JavaCycO

JavaCycO relies on the libunixdomainsocket.so libraray, which uses a UnixDomainSocket to communicate with Pathway Tools. Since UnixDomainSocket is only available on Unix-like systems, JavaCycO must be installed on a Unix-like operating system. JavaCycO must be installed on the same machine as the Pathway Tools installation.

1. Copy the CycTools_v0.1.0-beta.jar file to your preferred installation directory. It is recommended to install JavaCycO in the "ptools-local" directory created by Pathway Tools.

2. Run the install_JavaCycO.sh file by double clicking the file and selecting "Run in Terminal".

    - Install location defaults to current folder, but can be changed by user during install.
    - If you have java installed in a location other than the default for Ubuntu, you will need to modify the installer script in a text editor to point to the location of the jni.h file.

The server portion of JavaCycO, the JavacycServer, can now be run by double clicking the runJavacycServer.sh file. This must be run in terminal mode.

Installing CycTools

CycTools is a stand-alone java application which can run on any operating system "orting java. Java 1.7 must be installed on the client machine in order to run CycTools. On Linux-type systems, you may have to set the permissions to allow CycTools.jar file to be executable by using chmod u+x on the file. Run CycTools either by double clicking the jar directly (may not work if jar files are not associated with java on your machine), double clicking the appropriate run_CycTools file (bat file for windows, sh file for linux) or with the terminal command:

- java jar CycTools.jar

In order to successfully connect to a JavaCycO instance, Pathway Tools must be running on a linux-like machine in API mode:

- ./pathway-tools api

  * From the directory pathway tools was installed to in step 2.2

The JavaCycO server must also be running on the same machine as Pathway Tools:

- ./runJavacycServer.sh

  * From the directory JavaCycO was installed to in step 2.3

CycTools Instructions

CycTools can connect to a local or remote instance of JavaCycO. Enter the IP address of the server (localhost if on the same machine) and the port (default 4444). The username and password are not necessary unless set up during the JavaCycO installation.

Frame Viewer

The frame viewer tool is used to inspect the data contents of a single frame in the PGDB. Frame names can be entered directly, or a substring search can be performed to look for frames with a given substring in their name.

Start by entering a frame ID or search term in the search box. Select the frame type that you would like to search for (exact frame ID matches will be returned regardless of type selected). Then press submit (or press enter in the search box). If any matches are found, a window will pop-up displaying the search results. If an exact frame ID match was found, it will be indicated in the top of the display window. Other frames which match by substring will be displayed below. Options are displayed with the frame ID in parenthesis followed by the display name of the frame (if one exists). Selecting a frame and pressing OK will load that frame in the frame view area.

Frames are displayed in simple ASCII text. The structure of the display is as follows: the name of the frame is displayed first. This is followed by a series of slot labels, one per line, with the number of values in that slot in parenthesis behind the slot label. Slot values for a given slot name are shown, one per line, indented once from their slot label. Some slot values have slot-value-annotations. If this is the case, on the line following a slot value will be two dashes

(–) followed by the annotation label, two indents in. Annotation values follow the annotation label.

After all slots, values, and annotations, the superclasses of the frame are printed, along with the JavaCycO object type that was associated with this frame and the database that the frame was loaded from.

The frame viewer can display any frame in the PGDB, including instance and class frames. Most slot and annotation labels exist as frames which can be viewed.

BioCyc Import

The BioCyc Import tool is designed to make importing spreadsheet data into frame objects in the PGDB easy. The import utility takes as input a spreadsheet formatted file of data, maps the data to frames in the PGDB, previews the resulting changes to the PGDB, and performs the update of the PGDB with the spreadsheet data.

Select Database and Import Type

Import types include

1. Slot Value Import
2. Annotation Value Import
3. GO Term Import
4. Create Transcriptional Regulation
5. Delete Frame and Dependents

For the slot value option, the first column must be the exact frame ID of the frame to be modified. The following column headers must match the slot labels of the frame to be changed. Values in that column represent slot values (and can be separated by the multiple value delimiter if multiple values are desired). As many columns as desired can be imported at once. Only one frame object is allowed per line, a single frame can have multiple rows of updates.

For the annotation value option, the first column must again be the exact frame ID of the frame to be modified. The second column must be the slot label of the slot to be annotated. The values in this column are the values to be annotated. Note that you cannot use multiple values here, or it would not be possible to know what value is to be annotated.

The GO Term option uses the format ID, GO Term, PubMedID, EVCode, Time Stamp (mm-dd-yyyy hh-mm-ss), Curator. This option will trigger Pathway Tools to automatically import additional GO Term information for any GO annotations imported. The time stamp will be converted to Lisp date time format before importing.

The Create Transcriptional Regulation option uses the format Regulator, Regulatee, Mode. It will create new regulation instances in setting the regulator as having a transcriptional regulatory effect on the regulate of the type specified in mode ("-" for downregulation, "+" for upregulation). The regulator and regulatee IDs must be valid internal identifiers in the database (alternate identifiers cannot be used). As such, the alternate identifier search feature is skipped when performing this type of import. The new regulation frames will be assigned unique sequential identifiers automatically.

The Delete Frame option only requires the ID of the frames to be deleted in the first column. This frame and all dependents will be deleted from the database. If the frames to be deleted are gene frames, then all protein products and associated enzyme reactions will be deleted.

Select Options

Select the data file which is to be uploaded to the PGDB.

1. Select Import Type: when using the Slot Value, Annotation Value import, or Delete Frame import, this should by the frame type that is being updated. When using GO Term import, this should be set to Proteins (where GO annotations are stored). When using the Create Transcriptional Regulation import, this setting is ignored. This setting is used in the search feature to narrow search results when matching an alternate identifier to frames in the database.

2. Select Input File: select the data file to import

3. Select File Format: the spreadsheet can either be comma-separated values or tab-separated values

4. Multiple Value Delimiter: if multiple values are to be added to a single slot, they can be separated with the delimiter given here. The delimiter is interpreted literally (non-regex) and defaults to "$". If multiple value imports are not used, this is optional.

5. Append new data to existing values?: If selected, the new data imported will be added to the existing data in the PGDB. If not selected, the existing data will be replaced entirely with the new data. *Only slots explicitly being imported will be affected*

6. Ignore Duplicates: if selected, CycTools will first check to see if the exact value already is contained by the frame. If so, it will not be added again. Especially useful to check this option if running the same import file multiple times.

7. Update Author Credits: this option can be used to assign one individual and/or one organization to give curator credit to when importing changes. The individual or organization must already exist in the database, and will be added to the "Credits" slot as having "Revised" the frame. This should not be set when using the Delete Frame import.

File Preview

The second step is the File Preview step. The contents of the file will be loaded to the screen for review. Ensure that the file contains the correct information and that the file is being loaded correctly (comma/tab separated). Note that the multiple value delimiter does not affect preview at this stage. Pressing preview will go to the next screen. This step may take several minutes, as every frame being updated must first be downloaded.

Preview Changes

The third step is to preview the changes to the database. At this stage, no changes have been made to the data in the database. The list on the top shows all frames which are being updated as per the spreadsheet data. Selecting a frame will call up the original frame data in the bottom left text area, and the modified data in the bottom right text area. All changes will be highlighted in the text to assist in viewing the differences. Additionally, the next diff button can be used to scan through the updates made to the data step by step. Be sure to verify that the changes to the database are the changes intended. A check box is provided that will filter out any frames from the list that do not result in modified data after the import. This can be useful if the same import data is used multiple times. If the user is satisfied with the proposed changes, the update database button can be pressed, which will perform the import and modify the database. This step may take several minutes.

Commit to Database

After the update is performed, the results of the update can be reviewed in the final screen. This will provide a log of the successful and failed imports. Use this information to verify the success of the import, or to track down problem data. Each individual import will be listed as either success or fail, will be timestamped, and will refer to the original row of data in the spreadsheet which that update represents. Note that it may be possible to have several updates refer to the same row of data.

At this point, the database is in a modified but unsaved state. If the user is satisfied with the update, the save button will save the changes. Otherwise, the cancel button will undo all changes to the database. The user also has the option of saving the change log to file.

Remember:

1. Always save copy of database before making changes.

2. Make sure database is unmodified at start of the import process.

3. Do not have multiple users accessing the database at this time (either through CycTools or Pathway Tools).

Using iPlant Atmosphere Virtual Machines

Running a Server on an Atmosphere Virtual Machine

Atmosphere provides computing resources to academic users in the form of virtual machines which can be launched and administered by the user. This provides an ideal setting for testing CycTools without having to set up a linux-based Pathway Tools installation on a local machine. Atmosphere virtual machines can be set up to minimize the effort needed to run user software

Launching an Atmosphere Virtual Machine

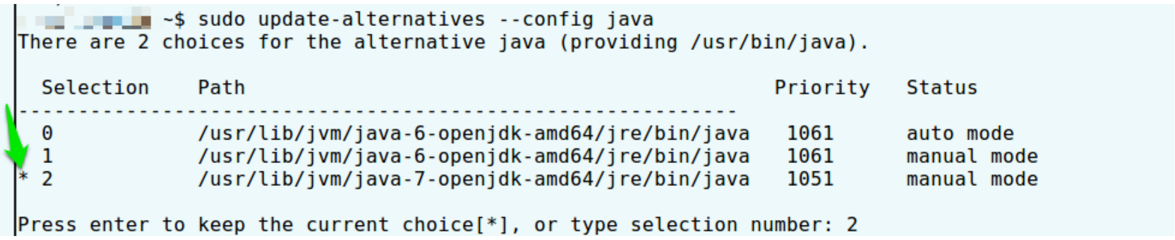
After creating an account at http://www.iplantcollaborative.org/ and logging in to the Atmosphere service https://atmo.iplantcollaborative.org/login/, you can create a virtual machine on which to run CycTools. Be sure to launch atmosphere at https://atmo.iplantcollaborative.org/application.

Launch a new instance using any suitable linux-based operating system. (CycTools was tested using Ubuntu 12.04 Unity GUI v1 for this guide). Once the machine has launched, access to the machine can be gained through several methods, including the web shell, VNC (for supported virtual machines), or through a ssh or terminal program such as putty (windows).

Log into the machine using your atmosphere username and password. You will need terminal (command line) access to the machine to continue.

Connecting through a terminal of your local machine, use the following command, replacing with your username and your virtual machines IP address:

- ssh -X username@123.456.789.10

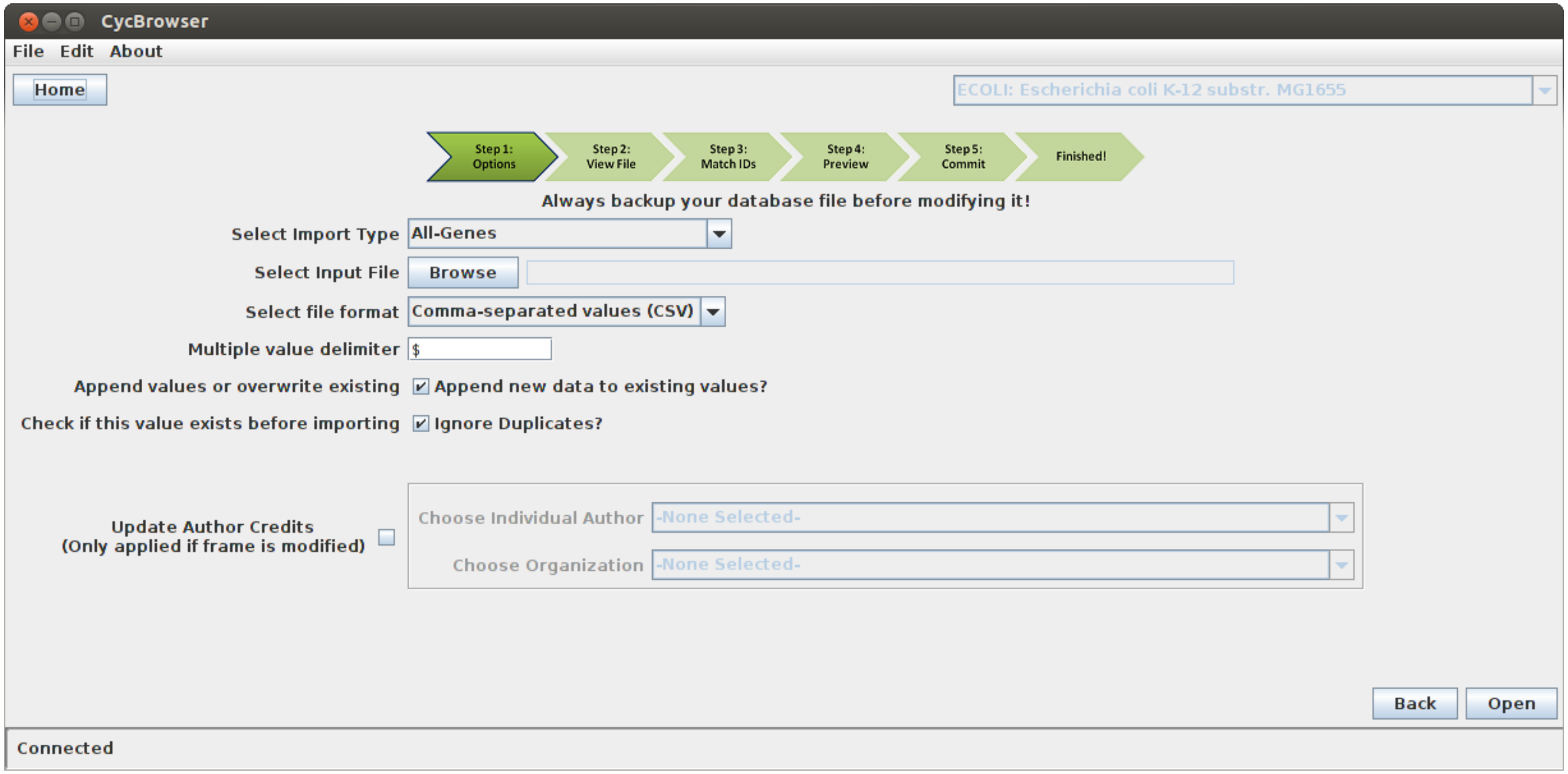


```
[redacted] ~$ sudo update-alternatives --config java
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                            Priority   Status
------------------------------------------------------------
  0            /usr/lib/jvm/java-6-openjdk-amd64/jre/bin/java   1061      auto mode
  1            /usr/lib/jvm/java-6-openjdk-amd64/jre/bin/java   1061      manual mode
* 2            /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java   1051      manual mode

Press enter to keep the current choice[*], or type selection number: 2
```

Figure A.1 Selecting Java version from installed versions.



Always backup your database file before modifying it!

Select Import Type: All-Genes

Select Input File

Select file format: Comma-separated values (CSV)

Multiple value delimiter: $

Append values or overwrite existing: ☑ Append new data to existing values?

Check if this value exists before importing: ☑ Ignore Duplicates?

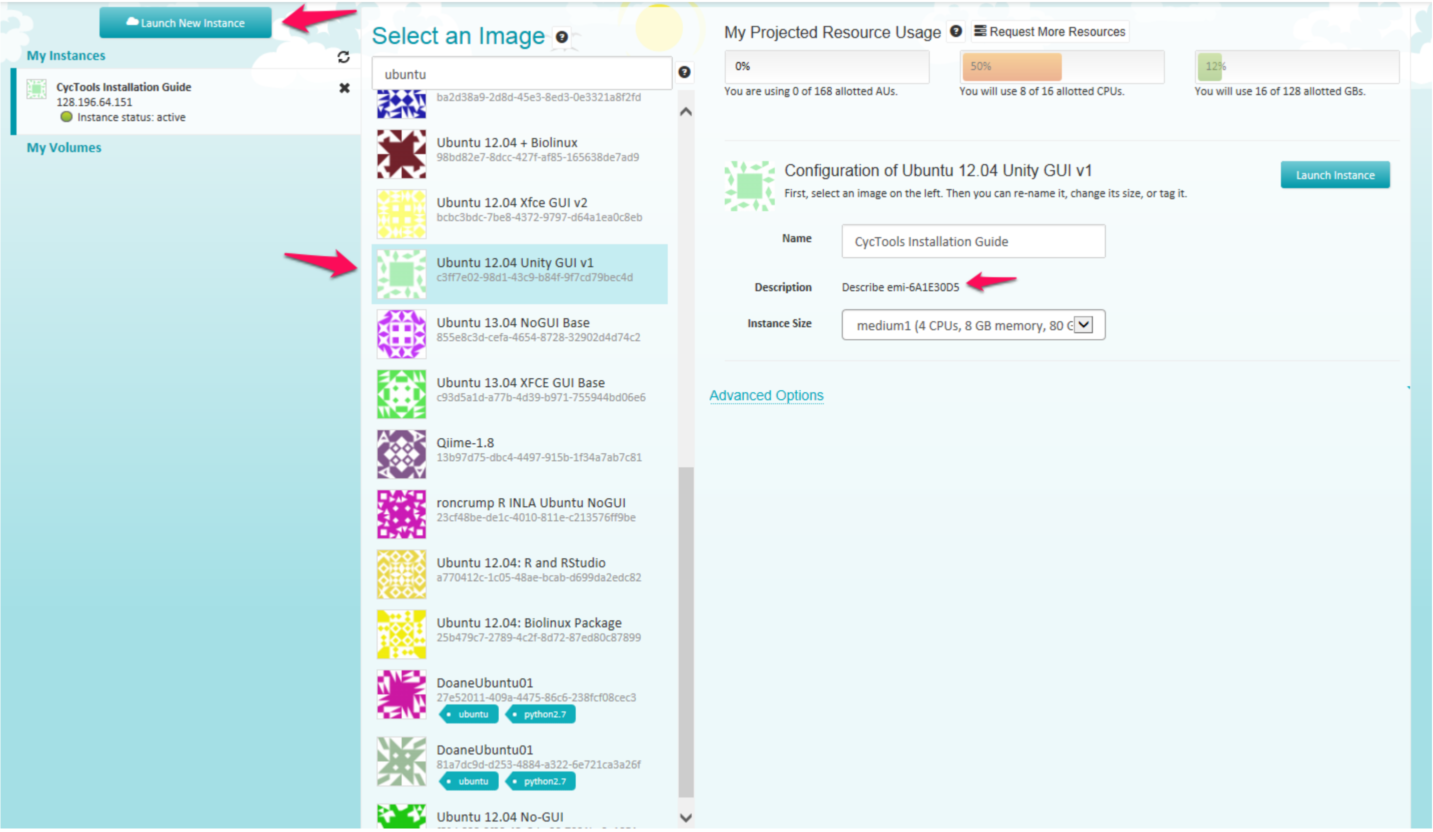
Update Author Credits (Only applied if frame is modified) □

Choose Individual Author: -None Selected-

Choose Organization: -None Selected-

Connected

Figure A.2 CycTools import options.



CycTools Installation Guide
128.196.64.151
Instance status: active

ubuntu

ba2d38a9-2d8d-45e3-8ed3-0e3321a8f2fd

Ubuntu 12.04 + Biolinux
98bd82e7-8dcc-427f-af85-165638de7ad9

Ubuntu 12.04 Xfce GUI v2
bcbc3bdc-7be8-4372-9797-d64a1ea0c8eb

Ubuntu 12.04 Unity GUI v1
c3ff7e02-98d1-43c9-b84f-9f7cd79bec4d

Ubuntu 13.04 NoGUI Base
855e8c3d-cefa-4654-8728-32902d4d74c2

Ubuntu 13.04 XFCE GUI Base
c93d5a1d-a77b-4d39-b971-755944bd06e6

Qiime-1.8
13b97d75-dbc4-4497-915b-1f34a7ab7c81

roncrump R INLA Ubuntu NoGUI
23cf48be-de1c-4010-811e-c213576ff9be

Ubuntu 12.04: R and RStudio
a770412c-1c05-48ae-bcab-d699da2edc82

Ubuntu 12.04: Biolinux Package
25b479c7-2789-4c2f-8d72-87ed80c87899

DoaneUbuntu01
27e52011-409a-4475-86c6-238fcf08cec3

DoaneUbuntu01
81a7dc9d-d253-4884-a322-6e721ca3a26f

Ubuntu 12.04 No-GUI

You are using 0 of 168 allotted AUs.

You will use 8 of 16 allotted CPUs.

You will use 16 of 128 allotted GBs.

Configuration of Ubuntu 12.04 Unity GUI v1

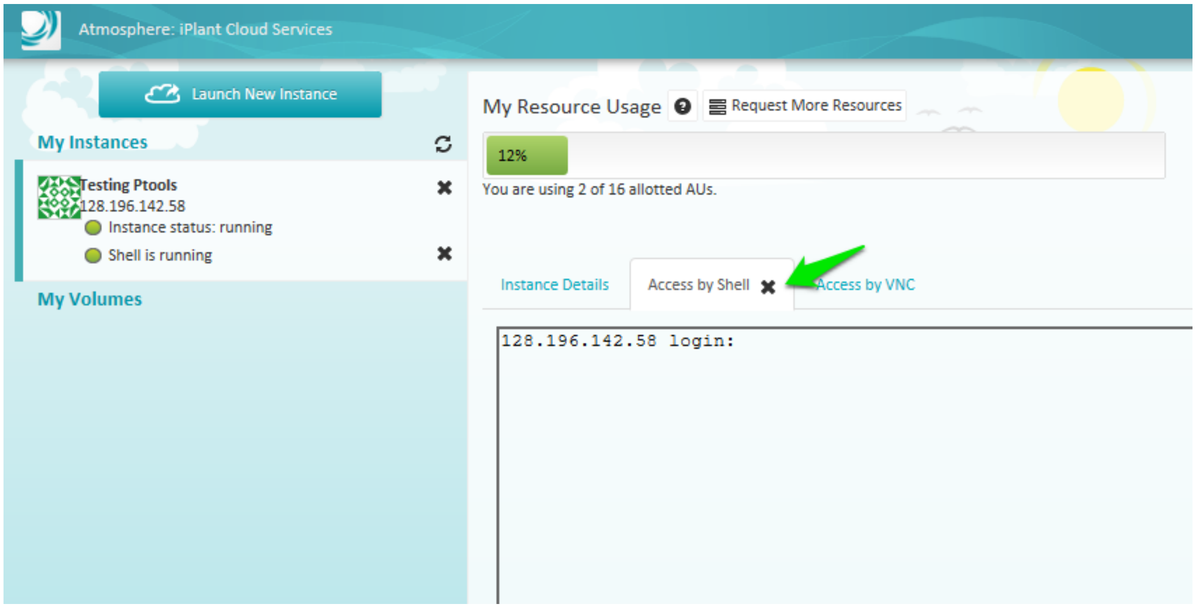
First, select an image on the left. Then you can re-name it, change its size, or tag it.

Name: CycTools Installation Guide

Description: Describe emi-6A1E30D5

Instance Size: medium1 (4 CPUs, 8 GB memory, 80 G

Figure A.3 Atmosphere home screen, launching a machine instance.



Testing Ptools
128.196.142.58
Instance status: running
Shell is running

You are using 2 of 16 allotted AUs.

```
128.196.142.58 login:
```

Figure A.4 Web based shell access.

APPENDIX B. SUPPLEMENTAL MATERIALS

Methods Used for Comparing CornCyc and MaizeCyc in Chapter 3.

Database Schema Structure

A Pathway Tools-based BioCyc database is organized as a collection of frames. A frame stores information representing either a single biological entity such as a metabolite or a gene, or a biological interaction such as a regulatory event or a pathway. Frames have named properties called slots, which describe the object they represent, such as the name of a gene, the molecular structure of a metabolite, or the reactions in a pathway. Frames are organized within the BioCyc database using the Pathway Tools ontology. The root of the Pathway Tools ontology is the frame "Things". The frame "Things" has only one child, the frame "Frames." Below "Frames" the ontology branches into major divisions, including biological entities such as "Chemicals" and "Enzymatic-Reactions", as well as metadata and annotation data such as "Databases", "People", and "Publications." The ontology is structured such that every frame (except the root frame "Things") can have one or more parent frames and zero or more child frames.

Within the frame ontology, a frame can represent either a class of frames or an instance frame. Instance frames contain specific information about biological objects, such as a particular gene or a specific chemical compound. Class frames group similar instances together and describe the general properties of the group. Class frames such as "All Genes" or "Pathways" define the properties that genes or pathways should specify. The class frames serve multiple purposes, including organizing the data within the resource so that it can be more easily found and referenced by both Pathway Tools and users, providing internal documentation describing what objects are represented by that class, and serving as a template for creating new frames of that class type.

Data Structure Comparison

Classes represent the core database structure of all BioCyc databases. They are created automatically during the database generation process by Pathologic as part of the Pathway Tools software, which selectively imports them from MetaCyc database based on enzymatic function assignments. Since classes are imported on an as-needed basis, the inclusion or exclusion of certain class frames can be an indication of differences in genomic and metabolic representation between two databases. If a class frame appears in two databases generated with the same Pathway Tools version, the content of the class frame is not expected to differ other than the changes made manually. The class information is updated only when with MetaCyc is updated. Table B.1 shows where the class structure differs between CornCyc and MaizeCyc. The classes unique to each database are listed in Tables B.3-B.7.

Gene and Protein Comparison

The Pathway Tools ontology defines several types of genes, but for our purposes we ignore Phantom-Genes and Pseudo-Genes categories in the schema and consider only genes classified under the "Genes" class. The red boxes in Figure B.1 represent Pathway Tools ontology class frames. Class frames in CornCyc and MaizeCyc have very similar content. Figure B.1 shows the ontology structure starting at "Genes". The red boxes (class frames) are identical in both CornCyc and MaizeCyc, showing the similarity in their organization. However, the blue boxes (instance frames) are not only labeled differently, they are placed in different locations in the Pathway Tools ontology. CornCyc, for example, has several genes placed directly under the Genes class, while MaizeCyc has a larger number of genes placed under the Unclassified-Genes class. There are 214 unclassified genes in MaizeCyc, but only 4 in CornCyc, mainly due to differences in computational pipelines and, to a smaller extent, in manual curation.

MaizeCyc and CornCyc classify their data differently within the Pathway Tools Ontology. MaizeCyc includes all gene data under the "ORF" category, while CornCyc includes most of its genes under the "ORF" category and some under the "Genes" category (see Figure B.1). This suggests that the most appropriate ontology category to compare gene data is at the "Genes" class and below. In CornCyc, 43 genes are stored directly under the “Genes” category. These are listed in Table B.2.

The text in the boxes of Figure B.1 represents internal identifiers for the frames in CornCyc and MaizeCyc. CornCyc internally assigned GDQC prefixes for genes, while MaizeCyc assigned GBWI prefixes for the same genes. Therefore, we are unable to match genes between CornCyc and MaizeCyc using frame id. Since the genes in both databases were annotated with their gene model names and transcript number suffixes, we are able to use a modified synonym search to match genes. For each gene, the gene model name was identified and the transcript-specific suffix stripped from the name. The resulting gene model names were matched between CornCyc and MaizeCyc.

CornCyc represents multiple splice variants for genes, while MaizeCyc only stores a single canonical transcript per gene for 99.5% of the genes it contains. Since Pathway Tools does not specify a standard format for storing transcript data, the transcript information is stored in gene objects using gene names suffixed with either a “_P##” (for protein) or “_T##” (for transcript), where ## represents a two-digit number to identify a given transcript. When matching transcripts, we homogenized the names by ensuring that all transcripts used the “T” suffix instead of the “P” suffix.

Reaction and Compound Comparison

Common names of reactions are not consistent between both resources. This can be due to either typographical differences or missing information either resource. Approximately 63 reactions were not given a descriptive common name in CornCyc but were given one in MaizeCyc, and 66 reactions were not given a descriptive name in MaizeCyc but were given one in CornCyc. Examples of typographical differences include the use of special characters (beta-carotene 3-hydroxylase vs. β-carotene 3-hydroxylase), formatting markup (tryptophan<em>N</em>-monooxygenase vs. tryptophan N-monooxygenase), and equivalent names (2-oxo-3-phenylpropanoate dioxygenase vs. phenylpyruvate dioxygenase). Due to naming convention differences, we had greater success matching based on frame IDs rather than reaction names.

Compounds are imported from MetaCyc during the early database creation steps of Pathologic on an as-needed basis. Since both CornCyc and MaizeCyc imported compounds from MetaCyc, we were able to match them based on frame ID's. For each compound matched in this way, we checked to see if the compound names and InChI strings matched. InChI strings are designed to facilitate computational representations of chemical compounds, therefore InChI matches should verify that the compounds have the same structure for more accurate matching. We found that many compounds which matched based on frame ID did not match name or InChI string. This is likely due their use of different source versions of MetaCyc used during their creation.

Pathway Comparison

Pathways are imported from MetaCyc during the Pathologic inference steps based on the reaction complement of the database. Since both CornCyc and MaizeCyc imported pathways from MetaCyc, we were able to match pathways based on frame IDs. Superpathways were excluded from the matching step as they simply represent a collection of standard pathways.

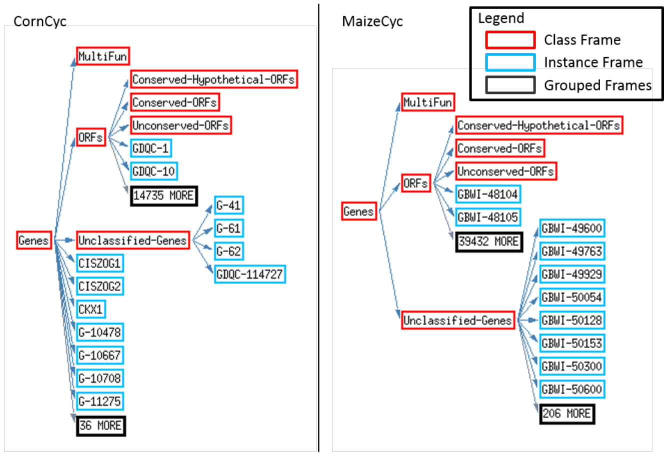


Figure B.1 An example of the Pathway Tools Ontology for (Left) CornCyc 4.0 and (Right) MaizeCyc 2.2. Red class frames represent the Pathway Tools Ontology structure, while blue instance frames represent information about Maize genes. The text represents the internal identifier for the genes. In addition to differences in information content between these two resources, the gene information is sometimes stored in different locations despite having the same basic Pathway Tools Ontology structure. The structure categorizes data content. MaizeCyc lists more genes as uncategorized. CornCyc includes 43 genes that are directly under the Genes category instead of the ORFs category.

Table B.1 The distribution of classes in CornCyc v4.0 and MaizeCyc v2.2. Although the Pathway Tools ontologies for CornCyc and MaizeCyc are very similar, there are slight differences in protein, compound, and pathway class presence between the two resources.

	CornCyc	**Overlap**	**MaizeCyc**
Gene Classes	0	247	0
Proteins Classes	13	377	32
Compound Classes	18	2,675	16
Reaction Classes	0	39	0
Pathway Classes	0	512	3

Table B.2 Genes listed as direct children of Genes class in Corncyc v4.0

FrameID	CommonName
CISZOG1	czog1
CISZOG2	czog2
CKX1	cko1
G-10478	sm2
G-10667	ZmDMAS1
G-10708	YS1
G-11275	pp
G-11284	LPE1
G-11369	tps23
G-11534	ADH1
G-12030	Z-ISO
G-12034	ZDS
G-12279	TPS6
G-14667	cpps2
G-14794	Bx7
G-1588	zpu1
G-3021	iaglu
G-3441	CHI_MAIZE
G-5541	OBT14DM
G-7161	MPAO
G-7822	Bx6
G-7841	Igl
G-7842	Bx1
G-7861	Bx2
G-7862	Bx3
G-7863	Bx4
G-7864	Bx5
G-8201	Bx8
G-8202	Bx9
G-8221	Glu1
G-8361	MIPS
G-8381	lpa3
G-8441	Ipk
G-9254	CKS
G-9834	TPS1
G-9837	TPS10
GDQC-114715	spi1
GDQC-114717	bx7
GDQC-114719	yuc1
GDQC-114720	na1
GDQC-114728	vp15
GDQC-119610	lox10
|me1|	me1

Table B.3 Protein classes unique to CornCyc

|kiwellin|
|mitochondrial-intermediate-protein|
|kissper|
|endothelin-1|
|Mitochondrial-Preproteins|
|RAD21-Cohesin-Subunits|
|Processed-Mitochondrial-Proteins|
|big-endothelin|
|Octapeptides|
|mature-protein|
|kith|
|Large-peptides|
|Small-peptides|

Table B.4 Protein classes unique to MaizeCyc

|Oxidized-Rusticyanins|
|helper-component-proteinease|
|Apo-AsbD-Proteins|
|picornavirus-polyprotein|
|togavirus-structural-polyprotein|
|type-IV-prepillin|
|fucosylated-protein|
|SoxZY-S-Thiocysteine-Sulfate|
LIMULUS-CLOTTING-FACTOR-B
|poliovirus-polyprotein|
|Non-lipoylated-domains|
|synaptobrevin|
|larger-subunit-of-tyrosine-aminotransfer|
|Cleaved-Synaptobrevin|
|pro-interleukin-1beta|
|Azurins|
|Reduced-Azurins|
|complement-subcomponent-C1s|
|EGF-domain|
|Lipoylated-domains|
|NPRS-Aryl-Carrier-Proteins|
|repressor-LexA|
|Sulfur-binding-protein|
|Oxidized-Azurins|
|Archaeal-Preflagellins|
|limulus-proclotting-enzyme|
|Potyvirus-Polyproteins|
|proacrosin|
|Reduced-Rusticyanins|
|Rusticyanins|
|flavivirus-polyprotein|
|Cleaved-togavirus-Struct-Polyproteins|

Table B.5 Compound classes unique to CornCyc

TRIPEPTIDES
|tRNA-with-ribothymidine-54|
|Piperdines|
CPD-14375
|Phytoalexins|
|Glucosyl-Cermaides|
|Anthocyanidin-3-O-sophorosides|
|Oligo-ADP-Rib|
|4-Hydroxyisoflavones|
|Coniferyl-Esters|
|N-acetyl-D-glucosamine-asparagine|
|Imidazoles|
|Organic-heteromonocyclic-compounds|
|a-diazole|
NADHX
|a-glycopeptide-D-mannosyl-Nsup4sup-N-ace|
|Folatepolyglutamate-n|
|4-Methoxyisoflavones|

Table B.6 Compound classes unique to MaizeCyc

14-BETA-D-XYLANS
CPD-1790
|7-hydroxyisoflavonoids|
SPHINGOSINE-CERAMIDES
14-ALPHA-D-GALACTURONIDE
|Cyanidin-3-O-rutinosides|
|D-xylooligosaccharides|
CPD1G-1530
|Anthocyanin-3-O-beta-D-glucosides|
|1-Acylglycero-Phosphocholines|
|Protein-3-phospho-L-histidines|
|DNA-Cytosine|
CPD-12999
|Glucosyl-Ceramides-II|
|Cyanidin-rutinoside-glucosides|
|Xyloglucans-Galactose|

Table B.7 Pathway classes unique to MaizeCyc

<table>
<tr><td>|UDP-Sugars|</td></tr>
<tr><td>CYCLITOLS</td></tr>
<tr><td>REDUCTANTS</td></tr>
</table>

APPENDIX C. LARGE TABLES AND FIGURES

Table C.1 Predicted change in regulatory states between wild-type and C6 fatty acid overproduction for 198 transcription factors in *E. coli* compared to bootstrapped randomized simulations. p-values calculated with Wilcoxon Rank Sum non-parametric test. Mean and standard deviation refer to the bootstrapped sample.

GeneName	TFName	Mean	StdDev	Percentile	Value	p-Value
yafQ	b0225	0.919656	3.619031	100	25.965993	0.088195
kdpE	b0694	1.621517	5.878051	100	31.38697	0.088195
rcdA	b0846	0.817841	4.799085	100	15.656963	0.088195
yeiL	b2163	0.165941	3.14029	100	21.007779	0.088195
yqhC	b3010	0.942437	3.931373	100	16.88873	0.088195
zntR	b3292	0.60279	4.164618	100	18.646093	0.088195
gadW	b3515	1.091868	4.663192	100	32.485643	0.088195
metR	b3828	0.538641	4.776521	100	26.799508	0.088195
ulaR	b4191	0.653071	4.48879	100	16.545332	0.088195
trpR	b4393	1.29439	4.489113	100	21.711904	0.088195
appY	b0564	0.488574	3.272473	98.387097	9.728861	0.099035
sdiA	b1916	0.586896	3.6786	98.387097	10.508827	0.099035
fhlA	b2731	0.618114	4.485686	98.387097	9.730605	0.099035
cspA	b3556	0.66976	4.602606	97.580645	10.520861	0.104841
gadX	b3516	2.416135	9.148602	96.774194	29.907534	0.110913
xylR	b3569	1.808196	3.712448	96.774194	10.055584	0.110913
ascG	b2714	0.248416	3.663328	95.967742	6.753869	0.11726
acrR	b0464	0.983779	4.511283	95.16129	10.00488	0.123889
lsrR	b1512	0.520542	4.511775	95.16129	8.786822	0.123889
rcnR	b2105	0.680593	4.308918	95.16129	8.62465	0.123889
cra	b0080	0.972719	4.967339	94.354839	10.366392	0.130806
cdaR	b0162	0.219743	4.571962	94.354839	7.463722	0.130806
fabR	b3963	0.239404	4.579339	94.354839	7.849774	0.130806
narP	b2193	1.303675	4.842723	93.548387	10.238224	0.138019
glcC	b2980	1.816376	4.380252	93.548387	10.426439	0.138019
nanR	b3226	0.516051	3.672247	93.548387	6.574348	0.138019
soxR	b4063	0.551179	2.883188	93.548387	5.180561	0.138019

Table C.1 (Continued)

GeneName	TFName	Mean	StdDev	Percentile	Value	p-Value
nemR	b1649	0.712302	3.965519	91.935484	8.69043	0.153361
fliZ	b1921	0.712365	3.584575	91.935484	5.77343	0.153361
atoC	b2220	0.554242	3.164078	91.935484	5.195283	0.153361
dcuR	b4124	0.70309	4.540258	91.129032	7.110462	0.161502
rclR	b0305	0.416424	3.013916	90.322581	5.663238	0.169966
lacI	b0345	0.421215	4.364873	89.516129	7.293177	0.178758
birA	b3973	0.243034	4.257245	88.709677	5.141066	0.187884
yehT	b2125	1.469128	4.302893	87.096774	6.105103	0.207158
rutR	b1013	-0.834348	11.370709	86.290323	7.130464	0.217316
ydeO	b1499	0.710657	3.963857	86.290323	5.740057	0.217316
uxuR	b4324	0.224848	5.559894	86.290323	5.577602	0.217316
uhpA	b3669	0.717141	4.410935	83.870968	2.924995	0.249928
xapR	b2405	0.829492	5.75291	83.064516	5.818024	0.261522
hyfR	b2491	1.664679	4.063123	83.064516	3.194799	0.261522
argR	b3237	0.649456	4.183308	83.064516	4.068615	0.261522
mqsA	b3021	1.425714	5.659237	80.645161	2.617498	0.298511
ebgR	b3075	1.080137	3.655438	79.032258	3.429317	0.325025
cynR	b0338	0.096447	4.255345	77.419355	2.789948	0.353025
malI	b1620	-0.050646	5.625932	76.612903	2.46605	0.367581
bluR	b1162	0.539474	4.249831	75	1.866868	0.397795
crp	b3357	2.418458	7.015185	72.580645	4.030936	0.445824
bolA	b0435	0.336684	4.102324	67.741935	-0.000056	0.551138
dinJ	b0226	1.109076	3.897682	66.935484	0.07428	0.56981
adiY	b4116	1.756053	4.830244	62.096774	0.595502	0.687691
mazF	b2782	0.134375	3.76605	61.290323	0	0.708206
hcaR	b2537	0.192781	4.247086	60.483871	-0.000174	0.728934
relB	b1564	-0.197256	4.824183	59.677419	-0.000304	0.749863
paaX	b1399	0.357723	3.961783	58.870968	-0.000108	0.770977
flhC	b1891	0.307521	4.68469	58.870968	0	0.770977
dicA	b1570	-0.063913	4.962645	57.258065	-0.001645	0.813704
alaS	b2697	0.482968	4.784174	57.258065	-0.000173	0.813704
purR	b1658	0.946287	4.004312	54.83871	-0.000144	0.878809
fur	b0683	0.408054	4.025711	54.032258	-0.000167	0.900717
galR	b2837	0.92883	6.596679	54.032258	0	0.900717
rhaS	b3905	-0.048093	5.189086	54.032258	-0.000583	0.900717

Table C.1 (Continued)

GeneName	TFName	Mean	StdDev	Percentile	Value	p-Value
feaR	b1384	1.032456	3.872958	53.225806	-0.000036	0.922701
ihfA	b1712	0.320073	4.385945	53.225806	-0.000239	0.922701
mlrA	b2127	0.657176	2.859202	53.225806	-0.000078	0.922701
nadR	b4390	0.22827	3.291414	51.612903	-0.000925	0.966829
phoP	b1130	-0.619036	4.741953	50.806452	-0.000303	0.98894
accB	b3255	1.517044	4.101309	50.806452	-0.000071	0.98894
nagC	b0676	0.416646	3.40218	50	-0.001012	1
hns	b1237	0.389189	6.694878	49.193548	0	0.98894
glpR	b3423	0.874641	5.141547	48.387097	-0.000249	0.966829
dnaA	b3702	0.834066	4.455244	47.580645	-0.000168	0.944744
hdfR	b4480	1.13788	5.028532	47.580645	-0.000605	0.944744
fnr	b1334	0.075841	3.610398	45.967742	-0.002616	0.900717
aidB	b4187	1.295289	4.609071	45.16129	-0.000855	0.878809
cbl	b1987	-0.548147	4.68527	42.741935	-0.002041	0.813704
cusR	b0571	0.173992	5.164836	41.935484	-0.001612	0.792263
lexA	b4043	0.170472	4.257978	41.129032	-0.001626	0.770977
gutM	b2706	0.103235	2.853801	40.322581	-0.009163	0.749863
pepA	b4260	-0.801884	6.09355	40.322581	-0.004629	0.749863
stpA	b2669	0.154854	4.36517	39.516129	-0.004993	0.728934
cpxR	b3912	1.620008	3.595818	39.516129	-0.001058	0.728934
dpiA	b0620	1.927723	7.578516	38.709677	-0.006886	0.708206
deoR	b0840	-0.402376	5.13186	38.709677	-0.004674	0.708206
tdcR	b3119	0.329075	4.316862	38.709677	-0.001282	0.708206
prpR	b0330	1.194768	5.756557	37.903226	-0.005438	0.687691
yefM	b2017	0.669356	6.580668	37.903226	-0.003843	0.687691
exuR	b3094	0.158165	4.148834	37.903226	-0.004797	0.687691
fis	b3261	0.185131	3.07042	37.903226	-0.00035	0.687691
arsR	b3501	1.033118	3.681809	37.903226	-0.001105	0.687691
rbsR	b3753	0.758011	3.548673	37.903226	-0.006734	0.687691
lrp	b0889	0.676262	3.61461	37.096774	-0.001788	0.667404
lrhA	b2289	0.383798	4.126458	37.096774	-0.006657	0.667404
nikR	b3481	-0.254175	3.866402	37.096774	-0.015716	0.667404
ypdB	b2381	-0.083937	4.377119	36.290323	-0.003692	0.647356
yqjI	b3071	0.229761	4.770469	36.290323	-0.036811	0.647356
idnR	b4264	0.672098	3.716308	36.290323	-0.001009	0.647356

Table C.1 (Continued)

GeneName	TFName	Mean	StdDev	Percentile	Value	p-Value
caiF	b0034	0.427132	5.290358	35.483871	-0.001098	0.627562
allR	b0506	0.086234	4.59684	35.483871	-0.000473	0.627562
rstA	b1608	0.708104	4.468501	35.483871	-0.007075	0.627562
kdgR	b1827	0.101524	3.140228	35.483871	-0.029703	0.627562
lysR	b2839	0.085932	4.770694	35.483871	-0.022785	0.627562
comR	b1111	1.018877	4.678022	34.677419	-0.031837	0.608032
cysB	b1275	0.382432	3.56923	34.677419	-0.002908	0.608032
rcsB	b2217	0.531676	4.910919	34.677419	-0.005997	0.608032
cytR	b3934	0.972223	4.344861	34.677419	-0.005092	0.608032
rob	b4396	0.157059	5.081176	34.677419	-0.006152	0.608032
betI	b0313	0.650113	5.256368	33.870968	-0.015722	0.588778
putA	b1014	1.247164	5.021114	33.870968	-0.005504	0.588778
fadR	b1187	0.874113	4.93287	33.870968	-0.002053	0.588778
gcvA	b2808	0.928578	3.20236	33.870968	-0.00219	0.588778
qseB	b3025	0.573898	5.394733	33.870968	-0.006243	0.588778
iclR	b4018	1.278492	4.152653	33.870968	-0.000845	0.588778
cueR	b0487	0.806312	4.759597	33.064516	-0.005801	0.56981
pgrR	b1328	-0.080529	3.969905	32.258065	-0.228832	0.551138
chbR	b1735	1.20529	4.099974	32.258065	-0.007834	0.551138
lldR	b3604	-0.672132	4.770775	31.451613	-0.304741	0.532773
metJ	b3938	1.290231	4.828648	31.451613	-0.002048	0.532773
tyrR	b1323	0.763857	4.081907	30.645161	-0.002153	0.514722
rcsA	b1951	0.973931	4.193846	30.645161	-0.008901	0.514722
glrR	b2554	1.009056	4.629046	30.645161	-0.003607	0.514722
mazE	b2783	0.578431	3.512233	30.645161	-0.006498	0.514722
glnG	b3868	1.145631	5.374847	30.645161	-0.027858	0.514722
murR	b2427	1.038577	3.592849	29.83871	-0.011745	0.496995
tdcA	b3118	1.534206	4.333921	29.83871	-0.003849	0.496995
rpiR	b4089	1.142205	4.512219	29.83871	-0.046364	0.496995
nhaR	b0020	1.16862	3.574749	29.032258	-0.001719	0.479598
torR	b0995	0.785239	4.013394	29.032258	-0.007795	0.479598
rtcR	b3422	1.082299	4.487101	29.032258	-0.00135	0.479598
araC	b0064	0.57865	4.395256	28.225806	-0.55016	0.462539
ada	b2213	1.211109	4.24066	28.225806	-0.011333	0.462539
mntR	b0817	0.742222	4.406387	27.419355	-0.002491	0.445824

Table C.1 (Continued)

GeneName	TFName	Mean	StdDev	Percentile	Value	p-Value
dhaR	b1201	-0.240369	5.448691	26.612903	-0.004259	0.429458
mlc	b1594	0.774611	6.428501	26.612903	-0.006134	0.429458
nac	b1988	2.297817	5.938084	25.806452	-0.003213	0.413447
sgrR	b0069	0.42957	4.262785	25	-0.104819	0.397795
argP	b2916	0.342141	4.882111	25	-0.036061	0.397795
ttdR	b3060	1.506688	5.409094	25	-0.003174	0.397795
agaR	b3131	1.731155	4.673198	25	-0.008551	0.397795
mcbR	b1450	-0.18693	7.573531	24.193548	-0.020894	0.382505
leuO	b0076	0.760143	4.201202	23.387097	-0.025136	0.367581
pdhR	b0113	1.192798	2.899522	23.387097	-0.002875	0.367581
ecpR	b0294	1.939033	4.508814	23.387097	-0.002624	0.367581
fldB	b2895	0.294166	3.57244	22.580645	-0.007316	0.353025
narL	b1221	0.938225	4.297437	21.774194	-0.022521	0.33884
marR	b1530	0.635695	3.670826	21.774194	-0.081893	0.33884
norR	b2709	2.150247	6.310322	21.774194	-0.126313	0.33884
malT	b3418	1.232262	4.772522	21.774194	-0.017945	0.33884
mtlR	b3601	0.466694	3.888206	21.774194	-0.009416	0.33884
zraR	b4004	1.066001	4.252393	21.774194	-0.002396	0.33884
envY	b0566	0.611948	4.033521	20.16129	-0.700258	0.311582
allS	b0504	0.91734	4.388762	19.354839	-0.010201	0.298511
mngR	b0730	1.381351	4.249192	19.354839	-0.005729	0.298511
csgD	b1040	0.327306	3.421979	19.354839	-2.144185	0.298511
hypT	b4327	1.499359	4.46169	18.548387	-0.005713	0.285812
mprA	b2684	0.520725	3.598116	17.741935	-0.036376	0.273482
rhaR	b3906	-0.238405	4.915264	16.935484	-4.228046	0.261522
hipB	b1508	0.986167	2.910174	16.129032	-0.014856	0.249928
srlR	b2707	1.324201	3.451051	15.322581	-0.00351	0.238698
dsdC	b2364	1.834424	4.238244	14.516129	-0.00825	0.227828
modE	b0761	-0.580469	10.400814	13.709677	-1.18089	0.217316
hipA	b1507	1.16265	3.336948	11.290323	-0.142533	0.187884
ompR	b3405	1.075542	4.409475	10.483871	-4.565284	0.178758
gntR	b3438	1.374321	4.400577	8.870968	-3.183695	0.161502
nrdR	b0413	-0.405302	4.823164	8.064516	-7.874614	0.153361
zur	b4046	0.94573	4.690901	8.064516	-3.910839	0.153361
phoR	b0400	1.102055	3.748464	7.258065	-3.145146	0.145535

Table C.1 (Continued)

GeneName	TFName	Mean	StdDev	Percentile	Value	p-Value
envR	b3264	1.196468	4.482216	5.645161	-5.661052	0.130806
relE	b1563	1.167041	5.021102	4.83871	-7.749675	0.123889
puuR	b1299	0.889007	4.066593	4.032258	-5.011861	0.11726
pspF	b1303	1.057475	3.792235	4.032258	-4.952479	0.11726
evgA	b2369	0.352683	5.197896	4.032258	-15.110033	0.11726
soxS	b4062	0.204607	3.817785	4.032258	-8.289854	0.11726
yoeB	b4539	1.034568	4.760594	3.225806	-10.171796	0.110913
iscR	b2531	1.67281	6.208229	1.612903	-6.317983	0.099035
yiaJ	b3574	1.731902	5.962735	1.612903	-4.871243	0.099035
oxyR	b3961	0.721954	4.392949	1.612903	-9.944534	0.099035
basR	b4113	1.205515	4.548228	1.612903	-7.076171	0.099035
mhpR	b0346	1.109807	4.747093	0.806452	-8.156068	0.093489
hupB	b0440	1.142156	5.098686	0.806452	-12.105485	0.093489
marA	b1531	0.359241	3.295065	0.806452	-10.824022	0.093489
slyA	b1642	0.41485	4.122021	0.806452	-8.216641	0.093489
baeR	b2079	0.963827	4.525735	0.806452	-12.211249	0.093489
creB	b4398	1.42335	5.634084	0.806452	-11.827355	0.093489
phoB	b0399	0.833039	4.150607	0	-9.254769	0.088195
ihfB	b0912	1.315517	3.88809	0	-56.700736	0.088195
uidR	b1618	1.463413	3.865873	0	-14.313008	0.088195
flhD	b1892	-0.091484	3.910091	0	-21.018625	0.088195
csiR	b2664	0.918404	4.42976	0	-17.353502	0.088195
gadE	b3512	0.465011	5.19784	0	-13.238294	0.088195
asnC	b3743	0.113147	4.194924	0	-23.125725	0.088195
ilvY	b3773	1.230951	4.708258	0	-14.177744	0.088195
melR	b4118	1.40598	5.05061	0	-16.270801	0.088195
cadC	b4133	1.398553	4.715246	0	-38.843936	0.088195
nsrR	b4178	1.285708	5.349467	0	-31.154859	0.088195
treR	b4241	1.22839	4.87084	0	-15.963621	0.088195
bglJ	b4366	1.298395	4.213126	0	-8.488729	0.088195
arcA	b4401	1.014951	4.57753	0	-24.578894	0.088195

,

```
(PYRUVATE NIL (
(OCELOT-GFP::PARENTS |2-Oxo-carboxylates|)
(NON-STANDARD-INCHI "InChI=1S/C3H4O3/c1-2(4)3(5)6/h1H3,(H,5,6)/p-1")
(ATOM-CHARGES (6 -1))
(INCHI "InChI=1S/C3H4O3/c1-2(4)3(5)6/h1H3,(H,5,6)/p-1")
(DBLINKS (CHEMSPIDER "96901" NIL |kothari| 3563632303 NIL NIL)
 (PUBCHEM "107735" NIL |taltman| 3466375285 NIL NIL)
 (KNAPSACK "C00001200" NIL |achi1| 3445698172 NIL NIL)
 (CHEBI "15361" NIL |taltman| 3452363604 NIL NIL)
 (LIGAND-CPD "C00022" NIL |kr| 3346617699 NIL NIL) (CAS "127-17-3")
 (UM-BBD-CPD "c0159" NIL |kawakami| 3278871244 NIL NIL))
(:CREATION-DATE 3107123668)
(SYNONYMS "alpha-ketopropionic acid" "BTS" "&alpha;-ketopropionic acid"
 "acetylformic acid" "pyroracemic acid" "2-oxopropanoic acid"
 "pyruvic acid" "2-oxopropanoate" "2-oxo-propionic acid")
(GIBBS-0 -114.9d0)
(APPEARS-IN-LEFT-SIDE-OF PEPDEPHOS-RXN PEPSYNTH-RXN)
(MOLECULAR-WEIGHT 87.055)
(DISPLAY-COORDS-2D (-10803 -2997) (-3700 1184) (3921 -2553)
 (-3700 9434) (3921 -10803) (10766 2035))
(STRUCTURE-BONDS (6 3 1) (5 3 2) (2 3 1) (4 2 2) (1 2 1))
(STRUCTURE-ATOMS C C C O O O)
(COMMON-NAME "pyruvate") )
((GIBBS-0 -114.9d0 CITATIONS "GibbsGroups97")))
```

Example C.1 An example of “Lisp-format” import formatting described in Chapter 2.