COMPUTATIONAL METHODS FOR RECONSTRUCTION AND ENHANCEMENT OF GENOME-SCALE METABOLIC NETWORKS

NGUYEN NAM NINH

NATIONAL UNIVERSITY OF SINGAPORE

2016

NUS

COMPUTATIONAL METHODS FOR RECONSTRUCTION AND ENHANCEMENT OF GENOME-SCALE METABOLIC NETWORKS

NGUYEN NAM NINH

(B.Eng. (Hons.) Southern Federal University, Russia)

A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SINGAPORE

2016

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Nguyen Nam Ninh

1 August 2016

Acknowledgements

Firstly, I would like to express my deepest gratitude to my supervisor, Prof Leong Hon Wai, for his continuous support, invaluable advice and timely encouragement throughout my PhD study. Under his excellent guidance, I have gained comprehensive knowledge and essential skills for my future success. Being his student and teaching assistant, I was really inspired and have learnt a lot from his teaching philosophy and enthusiasm. I also thank him for his financial support during the hard time. I am feeling lucky to have such a great advisor like him.

Beside my supervisor, I would like to thank the rest of my thesis committee, Prof Wong Limsoon and Prof Ken Sung Wing Kin, for their insightful comments and suggestions on my research. Their immense knowledge and experience have always motivated me attaining high quality work. My sincere gratitude also goes to Prof Vongsangnak Wanwipa for her keen collaboration and timely support. Her expertise in systems biology has helped me a lot for my research in this field. I also thank the anonymous reviewers who have given invaluable comments and suggestions to improve my works.

I would like to thank my colleagues and friends, Chong Ket Fah, Sriganesh Srihari, Zhang Melvin, Yu Shuzhi, Oh Shunhao, Lu Bingxin, Nguyen Phi Vu, and many others, for discussing their ideas during group meetings, and for having fun together all these days.

Last but not least, I am indebted to my family: my parents, my wife Ngo Thi Tu Anh, and my daughter. They always bring great pleasure to me after school time. Without their spiritual support, I would have done nothing.

Summary

Metabolic networks have many useful applications through systems biology and bioengineering. Such network of metabolic reactions can be reconstructed from genome by exploiting the gene-enzyme-reaction relationship. However, the reconstruction process is time consuming and intensively involves expert labour, yet producing incomplete networks with many gaps, i.e., reactions without enzyme-genes. This thesis aims to develop a computational pipeline to minimize time and manual effort for reconstructing high quality networks. Three problems were addressed, namely, gap filling, enzyme annotation, and network assembly.

The first problem is to sufficiently fill the gaps that remain after a network has been reconstructed. Previous methods used enzyme families to find gap candidates; they failed for poorly characterized families. In our indirect approach, we relied on *any* relevant homolog, such that no potential candidate is missed. Multiple function predictors were retrofitted and integrated. This ensemble method MeGaFiller can putatively fill 35% of gaps in several networks that previous methods failed.

The second problem is to reliably annotate enzymes with high accuracy and coverage, thus minimizing network reconstruction errors for later steps. We developed a novel bottom up method, called EnzDP, based on protein functional domain composition and calibrated HMM profiles. EnzDP built more than 4000 substrate-specific highly accurate enzyme profiles. It achieved a 94% accuracy in solid 5-fold cross validations, and outperformed many other alternatives.

The third problem is to quickly build a connected network with minimum number of gaps from a given set of annotated enzymes. We showed that this problem is NP-Hard, and

designed an approximation algorithm, called NetA. NetA predicts and then optimizes reference pathways by deleting non-evidence reactions while maintaining network connectivity. It was used to re-assemble a network for *Aspergillus oryzae*, which resulted in a network of 742 unique reactions in 119 pathways, in which 72.4% of reactions have enzyme genes identified.

From these methods, an automated pipeline can be achieved by running EnzDP to annotate enzymes, then using NetA to build a network, and finally applying MeGaFiller to fill gaps. This pipeline can serve as a useful automated tool that would help pushing further research on cellular metabolism and systems biology, as well as leading to many metabolic engineering applications

Contents

List of publications

1. Nam Ninh Nguyen, Wanwipa Vongsangnak, Hon Wai Leong: **Retrofitting Function Prediction Methods to Fill Gaps in Metabolic Networks**. *Poster representation at RECOMB,* 2013.

2. Nam Ninh Nguyen, Wanwipa Vongsangnak, Bairong Shen, Phi Vu Nguyen, Hon Wai Leong: **MeGaFiller: A Retrofitted Protein Function Predictor for Filling Gaps in Metabolic Networks**. *J Proteomics Bioinform*, 2014, S9: 003.

3. Nam Ninh Nguyen, Sriganesh Srihari, Hon Wai Leong, Ket Fah Chong: **EnzDP: improved enzyme annotation for metabolic network reconstruction based on domain composition profiles**. *J Bioinform Comput Biol*, 2015. **13**(5): p. 1543003.

4. Nam Ninh Nguyen, Hon Wai Leong: **Using gap fillers and enzyme function predictors for enhancing reconstructed metabolic networks.** *Proceeding of the 6th international conference on Computational Systems-Biology and Bioinformatics 2015 (CSBio2015)*, 2015.

List of tables

List of figures

Chapter 1

Introduction

1.1 Overview

1.1.1 Metabolic network and metabolic reconstruction

Inside a cell, a network of interconnected biochemical processes are continuously carrying on to perform fundamental cellular functions. These metabolic processes are series of metabolic reactions which are catalyzed by specific types of enzymes. Essentially, these enzymes are products of genes that the cell expresses to drive its desired capabilities. A metabolic network summaries this ultimate relationship among metabolic genes, proteins and reactions, which provides a systematic tool for investigating metabolism.

Increasingly, metabolic networks play an important role in systems biology. They help reveal fundamental questions on evolutionary history of species. In fact, many metabolic pathways are common in all life forms, thus studying them contributes enormously to our understanding of life. More usefully, metabolic networks have a wide range of application through systems biology and bioengineering, from medicine, to food/energy industry processing.

A metabolic network can be reconstructed from a genome, exploiting the specificity of the link between enzyme genes and metabolic reactions. The first step is to identify enzymatic functions coded in the genome. After that, a list of metabolic reactions associated with these enzyme is collected. Finally a network is reconstructed containing these reactions. However, these are just general steps; in reality, many complicated tasks with intensive human

involvement must be done. Nevertheless, in recent years, there has been an effort to reconstruct genome-scale metabolic networks of hundreds species, such as yeast [1, 2], fungi [3-6]. Each of these networks contains thousands of metabolic reactions, involves thousands of genes and metabolites.

1.1.2 Challenges and approaches

Despite that many methods have been developed [7, 8], most if not all of the reconstructed networks are still far from complete [9, 10]. The incompleteness is indicated by the existence of significant number of gaps [11, 12] in the reconstructed networks, as well as by the inconsistency between metabolic networks of the same species that reconstructed at different time. Recently, a protocol [11, 13] for building high quality genome-scale metabolic network has been proposed, which involves many of steps to be done. Expert manual validation is essential for network quality control, thus it is unavoidable from network reconstruction. However, the manual curation task takes a lot of time. These issues become a bottle neck for obtaining high quality metabolic networks. Essentially, the use of computational tools that are highly reliable can substantially reduce the time and effort. Therefore, effective and reliable reconstruction-aid tools are of high demand.

Filling gaps is the most time consuming task [11, 13]. Gaps still exist after gap filling has been performed. These gaps are difficult-to-fill, since no or very little evidence has been found to pinpoint any candidate. For example, in a version of *A. oryzae* network [3], 52 out 61 gaps do not have proper protein family homologs, thus there is no direct way to identify potential candidates. This can only be unravelled by pursuing indirect approach such as our MeGaFiller (in Chapter 3).

In fact, the origin of missing knowledge [11, 12] and inconsistency come from very first steps of network reconstruction. For example, genome annotation using regular bioinformatics tools may suffer from annotation errors [14-16] and coverage lost. These "bugs" propagate through the reconstruction pipeline that makes them difficult to completely resolve in later steps. In other words, candidate enzymes/reactions that were missed or wrongly included in

previous step hardly can be recovered in reconstruction stage. For example, after using MeGaFiller for filling gaps in a reconstructed network of *A. oryzae*, we found that MeGaFiller can supply more genes for existing reactions, as well as predicts new reactions that were not included in the network previously. This reveals that the annotation data in previous step is not complete, having missed many true candidate genes and reactions. Therefore, the enzyme classification procedure should be revised to get more reliable result.

The most challenging problem in enzyme classification is that, one enzyme family can have multiple different protein domains, and at the same time, one domain can be carried by different families, as shown in Figure 4.1, Chapter 4 as an example. The relationship between enzyme families and protein domains is not one to one, but many to many. Thus, classifying enzymes based on protein domains alone may give misleading results. We carefully analyzed these cases and came up with a scoring scheme called DEAS to score the association between enzyme families and *domain architecture*. This score is used as a weight for clustering procedure. Together with this, we developed a stringent enzyme classifier EnzDP based on calibrated newly-built Hidden Markov Models profiles of protein domain composition. The method achieves high enzyme coverage as well as high prediction accuracy.

After having identified a set of reliable enzymes, a list of associated reactions can be retrieved from reaction databases. Then it is ready to reassemble them into pathways and connect these pathways into a network. Reconstruction is not a one-time task; it is a multiple phase iterative process, in which any step may cause the whole process to be re-executed if any new data is available. Thus, the challenge here is how to quickly re-produce a network in the emergence of additional information. Another challenge is to *minimize* the number of gaps while maintaining network reachability. Given this constraint, the problem is NP-Hard. To tackle these challenges, we relaxed the requirement of minimum number of gaps. After that, we developed an approximation algorithm NetA, which runs in polynomial time. Following graph-theoretic pathway-based approach, NetA predicts and then optimizes metabolic pathways by deleting no-evidence reactions while preserving network connectivity. Essentially, a quick and effective network reconstruction pipeline was achieved by running NetA together

with EnzDP (for getting enzyme set) and MeGaFiller (for filling poorly characterized gaps). This serves as a combined tool for quickly rebuilding a draft network from scratch.

1.1.3 The three problems

Three problems are addressed in this thesis to improve network reconstruction process. The first problem is to accurately predict the set of enzymes from the genome and thus leading to an accurate set of metabolic reactions, and minimizing reconstruction errors. The second problem is to effectively fill the gaps after the network has been previously reconstructed. The third problem is to quickly assembly a network with minimum number of gaps. These problems can be addressed and solved separately from the common reconstruction pipeline, but methods developed for them can be used as a combined tool.

1.2 Research aims and scope

1.2.1 Aims

The overall aim of the research in this thesis is to develop reliable high through-put computational methods in order to minimize time and manual effort for reconstruction of high quality genome-scale metabolic networks. This thesis addresses three important problems that can be separated from the common reconstruction protocol, namely: filling gaps in reconstructed networks, enzyme function annotation, and network assembly. The specific aims are:

- *To develop a novel and effective metabolic gap filler. The method would fill gaps in reconstructed networks, thus improving network quality.*
- *To develop novel enzyme classification methods with high enzyme coverage and high accuracy. The high accuracy and coverage would help minimizing metabolic gaps and/or errors in network reconstruction.*
- *To develop a quick automated network assembly method with minimized manual effort and missing knowledge. Together with the above two methods, a high-through put reconstruction pipeline is built.*

1.2.2 Research scope

Research scope mainly focuses on developing computational methods for minimizing manual effort in network reconstruction. Notably, the expert manual curation is unavoidable. The reconstruction protocol includes verification phase that involves in-silico modelling and experimental verification. These tasks should be done by expert systems biologists.

To check the prediction results made by the developed methods, manual validation using reference databases is performed. However, the main scope of the research focuses on computational aspect of network reconstruction. Specifically, for gap filling problem, the metabolic gaps in a reconstructed network will be addressed. Filling these gaps is the most challenging task after reconstructing a network, thus is an important part described in this thesis. On the other hand, the problem of filling the global gaps (i.e., gaps that are involved in all known species) requires a much large cross-species scale investigation, thus is out of current research scope. For enzyme function prediction, our work will focus on developing methods for increasing enzyme coverage at a considerable good accuracy, since enzyme coverage is more essential for minimizing reconstruction errors. For network assembly problem, computational approach produces initial network models; thus, further experimental verifications need to be done by experts to finalize these models.

1.2.3 Thesis outline

The outline of this thesis is following. This first chapter gives an overview on the problems addressing in this thesis, as well as research aims and scope. The second chapter introduces background and a literature review on general computational methods for reconstructing and enhancing metabolic networks. The third chapter presents our method MeGaFiller on filling critical gaps using retrofitted function prediction tools. Next chapter focuses on EnzDP - our novel enzyme classification method. After that, the fifth chapter describes our NetA method for assembling metabolic network, in which the enzyme annotation made by EnzDP is used. The last chapter gives conclusions on current research as well as addresses potential continuing research directions.

Chapter 2
Background

2.1 Metabolic networks and computational reconstruction methods

2.1.1 Metabolic network

Metabolic network of a cell is a mathematical model [17] describing all metabolic processes that happen inside the cell. These processes determine the cell's physiological and biochemical properties. They allow the cell to perform basic functions of life, such as growing, reproducing, maintaining cellular structures and responding to the outer environment. In fact, metabolism involves all processes that convert or use energy for living activities. More specifically, catabolic processes break down organic matter into basic blocks and harvest energy (e.g. in cellular respiration). On the other hand, anabolic processes use energy to build up all the cell components from those basic blocks (e.g. synthesis of proteins or nucleic acids).

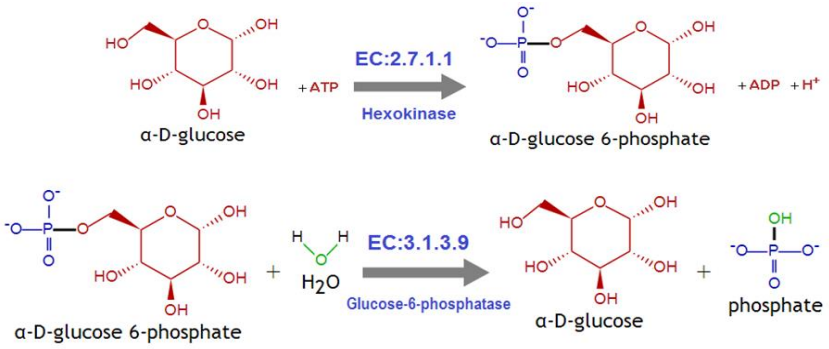

Figure 2.1 – Example of metabolic reactions.

The reaction on the top transforms *glucose* into *glucose-6-phosphate*, with help of *hexokinase* enzyme (EC:2.7.1.1). The reaction on bottom transforms *glucose-6-phosphate* back into *glucose*, with help of *glucose-6-phosphatase* enzyme (EC:3.1.3.9).

Metabolic network can be described as a network of cellular **metabolic reactions**. Such a reaction involves a transformation of metabolites (from substrates into products, e.g. Figure 2.1), with help of an enzyme catalyst. Ultimately, most of metabolic reactions do not occur spontaneously without catalyzed enzymes. These enzymes, which are products of genes, can lower the reaction energy barrier via theirs active sites, thus catalyze the transformation. With the presence of the specific enzymes, metabolic reactions occur much more effectively, with a speedup rate up to million times faster. Significantly, each enzyme has unique active sites that can catalyze for only a specific type of reactions where it is able to bind to the substrates. Enzyme specificity is the mechanism for a living organism to drive its desired metabolic reactions in an organized manner via its gene products. Different cell has different set of metabolic reactions, depends on the set of enzymes that are expressed in the cell's genome. This ultimate property suggests that, the presence of an enzyme in the genome infers the presence of the corresponding metabolic reaction in the cell, and vice versa.

Enzymes play an important role in cellular metabolic activities, and thus are also of the most important component of metabolic network. To classify enzymes and metabolic reactions, an enzyme nomenclature [18] is used. In this system, an enzyme commission number (**EC number**) is described by a sequence of 4 digits with format EC:x.x.x.x. The first digit describes the class of the reaction; the second and third digits describe sub and sub-sub class of the reaction, respectively; while the last digit describes the specific enzyme or substrate binding.

For example, the hexokinase enzyme is classified as EC:2.7.1.1, which means that, it is a transferase (EC:2.-.-.-) that transfers phosphorous-containing groups (EC:2.7.-.-), with an alcohol group as acceptor (EC:2.7.1.-). Enzymes are divided into 6 main classes as shown in Table 2.1. Ultimately, biochemical functions of enzymes and enzymatic reactions can be described via EC numbers.

Table 2.1 – Six classes of enzymes

EC number	Class	Function
EC 1.x.x.x	*Oxidoreductases*	Catalyze oxidation/reduction reactions
EC 2.x.x.x	*Transferases*	Transfer a functional group (*e.g.* a methyl or phosphate group)
EC 3.x.x.x	*Hydrolases*	Catalyze the hydrolysis of various bonds
EC 4.x.x.x	*Lyases*	Cleave various bonds by means other than hydrolysis and oxidation
EC 5.x.x.x	*Isomerases*	Catalyze isomerization changes within a single molecule
EC 6.x.x.x	*Ligases*	Join two molecules with covalent bonds

On the other hand, cellular chemical reactions occur in chains, in which products of first reaction become substrates for subsequent reactions, and so on. These chains of reactions form **metabolic pathways**. Each pathway performs a basic biological function for the cell. For example, the glycolysis pathway performs oxidation of glucose to obtain energy (in form of ATP) and pyruvate (see Figure 2.2). This pathway is present in almost all organisms as a basic function of energy metabolism.

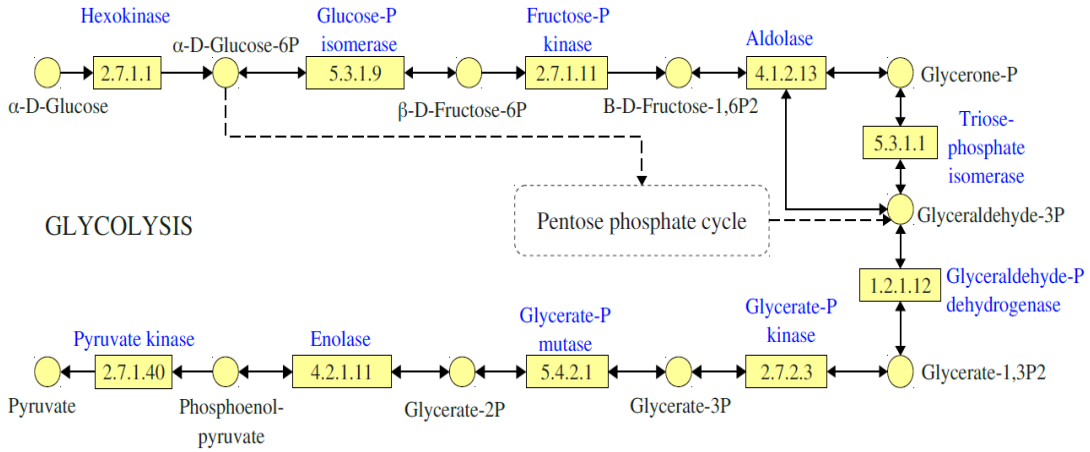

Figure 2.2 – Reactions in glycolysis pathway

Rectangles denote reactions with EC numbers. Small circles are metabolites. Arrows present the substrate-reaction relationship. Dashed arrows present the interconnection with other pathways (denoted by dashed rectangles).

For a more detailed view, metabolic network of a cell is a collection of all interconnected metabolic pathways that are present in the cell and containing all of its biochemical reactions. Such a network may contain hundreds of pathways with thousands of reactions. For example, a metabolic network reconstructed for *Aspergillus oryzae* [3] contains 2360 reactions, spread over 92 pathways.

Depends on investigation purposes, metabolic networks can be visualized by different **network representations**. A network can be represented by undirected/directed graph of metabolites/reactions; bipartite digraph of reactions and metabolites; or, a mixed model of digraph of reactions and metabolites together with undirected edges that describe enzyme-reaction relationship. In this thesis, we consider metabolic network as a directed graph of reactions and metabolites. In addition, the gene-protein-reaction relationship is described as a property of a reaction.

Metabolic network is one of the best known biological networks. Increasingly, metabolic networks play an important role in systems biology. It helps reveal fundamental questions on evolutionary history of species. In fact, many metabolic pathways are common in all life forms, thus studying them contributes enormously to our understanding of life. More usefully,

metabolic networks have a wide range of applications through systems biology and bioengineering, from medicine, to food/energy industry. For example, analyzing metabolic networks of pathogens and parasites can help identify effective drug targets in medicine. In food processing, metabolic models of microorganisms are analyzed to apply gene-engineering techniques for the purpose of achieving the desired products with maximum yields [19-22].

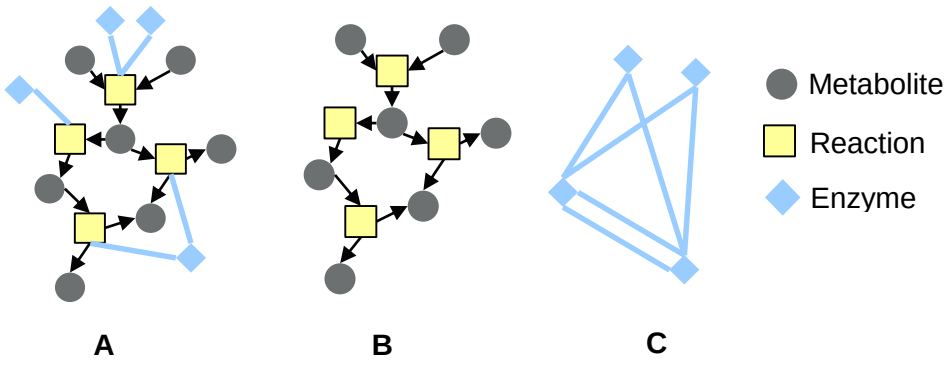

Figure 2.3 – Metabolic network representations

A: Network of reactions, metabolites and enzymes. B: Network of reactions and metabolites. C: Network of enzymes (aka enzyme gene association network)

In short, metabolic network summarizes knowledge about cell metabolism in a systematic way. Such a network contains thousands of metabolic reactions with their catalyzed enzymes. Enzymes are crucial for metabolic study, since they are products of genotypes yet are the "drivers" for phenotypes. The essential relationship between genes, proteins (enzymes) and metabolic reactions is visualized in metabolic networks. Identifying gene-protein-reaction relationship is a basic for reconstruction of a metabolic network from a genome.

2.1.2 Metabolic network reconstruction

Metabolic network of a microorganism can be reconstructed from its genome. A common **reconstruction pipeline** is illustrated in Figure 2.4, which consists of several tasks, from genome annotation, reaction identification, pathway assembly, to network verification [7, 8, 23]. This section describes general reconstruction pipeline, in which the problem of network reconstruction is often referred to one step in the pipeline, that is, pathway/network assembly.

From the input genome, the first step is to annotate the genome to collect metabolism-related data. This can be done with help of bioinformatics tools and reference annotation databases such as NR (*ftp://ftp.ncbi.nih.gov/blast/db/FASTA/*), UniProt [24, 25], Pfam [26, 27], CDD [28, 29], and KEGG [30-32]. In fact, genome annotation is also a starting point of any genomics investigation. This step can be separated from the pipeline as a single task. Thus, reconstruction often starts with annotated genome data.

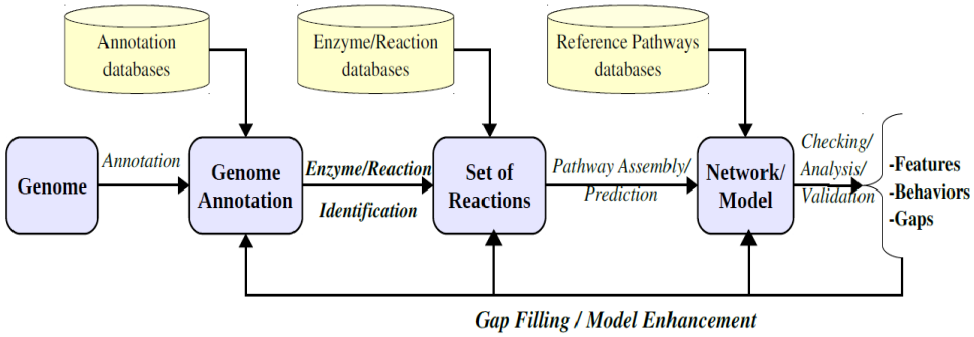

Figure 2.4 – Pipeline for reconstruction of metabolic networks from genomes

From annotation data, a set of enzymes is identified. This can be done by using mappings between functional annotation descriptors and EC numbers, such EC2GO [33], Pfam2GO [34], EC2PDB [35]. Recent function annotation methods can produce EC number prediction directly, such as PRIAM [36], EFICAz [37, 38] and ModEnzA [39]. Leveraging on the enzyme specificity assumption that presence of an enzyme infers presence of its catalyzing reaction, the set of metabolic reactions can be listed. All relevant information of these reactions can be retrieved from reference enzyme databases such as KEGG, BRENDA [40].

The next step is to identify the set of metabolic pathways that are present in the cell, using reference pathways database (KEGG, BioCyc [41]). The presence of a pathway can be inferred by the presence of a significant number of its reactions. After that, the set of identified pathways and reactions is assembled into a network. This draft network contains raw list of reactions and pathways.

Such a draft network can be converted to a stoichiometric metabolic model, by adding stoichiometric coefficients from reference reaction databases. From this point, network

checking/analysis can be performed to verify its consistency. After that, the model can be used for analyzing features of interest, as well as for useful applications

However, such a reconstructed network is just an initial version. After analyzing step, the network may contain inconsistency and gaps. It must be further refined and/or enhanced by filling gaps and revising previous reconstruction steps. The whole pipeline must be repeated several times until a consistent network model is achieved. Significantly, in most of the steps, expert curation is needed to control the quality of the reconstructing network. This effort often is the most time consuming [13], thus a quick network assemble tool is very useful. The problem of network assembly is presented in Chapter 5.

2.1.3 Gap in reconstructed networks

The comprehensiveness of reconstructed models greatly depends on quality of genome annotation. Although many sophisticated computational methods were developed to assist annotation process, many genes still remain with unknown functions, and other genes may be assigned incomplete or even incorrect functions. Those un/miss-classified genes may code for 20–60% of the proteins in most genomes [12]. For example, in the annotated genome of *Aspergillus oryzae* published in 2005 by Machida et al. [42], the number of hypothetical proteins is more than 50% of the annotated genes [3]. Furthermore, the metabolic databases used in reconstruction process may be incomplete, especially for less studied species. Hence, there are metabolic reactions that are not well characterized, and/or the enzymes catalyzed for these reactions are not known yet. The incompleteness of both metabolic databases and genome annotation is the original cause of missing metabolic knowledge, or the gaps, in reconstructed networks.

There are two types of gaps: missing reaction gaps and missing gene gaps. A missing **reaction gap** is the reaction that actually occurs inside the cell, but the reconstruction method has failed to include it into network, since there was no or very little known evidence to support its existence. These gaps can be indicated by the existence of dead-end metabolites, that is, metabolites that are consumed but not produced, or are produced but not consumed, by any

included reaction. Reaction gaps reflect errors in network reconstruction. Therefore, reconstruction methods try to minimize the number of reaction gaps.

The second type of gaps, missing gene gap (or **metabolic gap**), is a reaction that is included into the network but the gene encoding for the catalyzed enzyme is not known. Metabolic gaps are caused by genome annotation error/incompleteness and/or by failure of gap filling procedure that attempted previously. Filling these metabolic gaps are the most time consuming task in network refinement phase.

Figure 2.5 demonstrates an example of a metabolic gap in *A. oryzae* sub-network. The metabolic gap filling problem here is to find in *A. oryzae* genome the D-Xylose reductase genes.

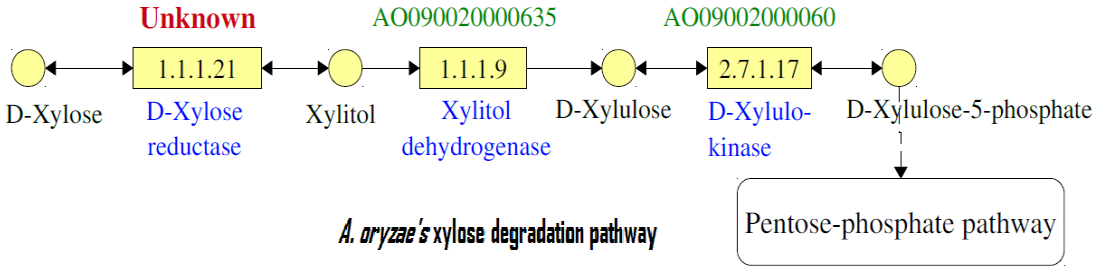

Figure 2.5 – An example of metabolic gap in *A. oryzae* metabolic network

This xylose degradation pathway contains 3 connected reactions. Two of them (EC:1.1.1.9 and EC:2.7.1.17) have already been found the enzyme coding genes (AO090020000635 and AO09002000060, respectively). The reaction with EC number EC:1.1.1.21 (D-Xylose reductase) which is a transformation between D-Xylose and Xylitol, is a gap, i.e. the gene encoding for D-Xylose reductase enzyme is unknown in the *A. oryzae* genome.

Metabolic gaps are further categorized into 2 types [12]: local gaps and global gaps. A **local metabolic gap** is a metabolic gap in the reconstructed network of a specific species of interest. However, it is not a gap for other species. This means that, the similar enzyme genes are identified in other species, but missed in the species of interest. On the other hand, a **global metabolic gap** is a gap for all known species. In other words, it is a globally missing gene that has not been found in any known genome yet.

A completely gap-free metabolic network must include all reactions that actually occur and each reaction must have support evidence at gene-level. This means that all the reaction gaps and the metabolic gaps must be filled. In general, resolving gaps consists of two steps

[17]: first, to identify reactions that could complete the network, and second, to find the missing enzyme genes. While the first task is mainly done in network reconstruction, the second task is usually separated from reconstruction pipeline as a single problem – the gap filling problem. Gap filling problem is presented in Chapter 3.

2.1.4 Enzyme function prediction

Genome annotation is a starting point of any genomics investigation. It is also the first step for metabolic network reconstruction. The specific genome annotation problem here is to predict metabolic functions for the given genome. This problem is called enzyme function prediction, or enzyme classification, which is a specific instance of function prediction problem.

Enzyme functions can be recognized via text description in reference databases (such as GeneBank, NR). However, this requires biologist-expert to interpret the text annotation. For a more convenient way, functions of enzymes are described via gene ontology (GO) terms or enzyme commission (EC) numbers. EC number precisely describes enzymatic activities of an enzyme, thus using EC numbers is the most preferred way.

If function annotation provides only GO term description, one may need EC2GO mappings [39] to map back GO terms to EC numbers. There are several other mappings that provide equivalence between different function descriptors, such as Pfam2GO [34], and ips2GO. These mappings can be used together with EC2GO to interpret EC number annotation.

Enzyme function prediction often uses genome sequences as the input. There also exist approaches that use protein-protein interactions or 3D structure comparison for inferring function. However, methods that use sequence input are of the most original and effective. In this research, we are interested in developing enzyme function prediction methods with high enzyme coverage and reliable accuracy to aid metabolic network reconstruction from the first step. Enzyme annotation is an important task to reduce gaps for later phase of gap filling and network refinement. Our enzyme function prediction method EnzDP is presented in Chapter 4.

2.2 Literature review

To thoroughly understand the three problems addressing in current work, literature reviews have been done. This chapter gives details on the metabolic gap filling problem, enzyme function prediction problem, and network assembly problem.

2.2.1 Enzyme Function Prediction Problem

Enzyme function prediction, or more generally, protein function prediction is a fundamental problem for any genomics investigation. Significantly, a huge amount of genomics data has been generated with an exponential rate, therefore analyzing and interpreting biological functions become challenging. Furthermore, experimentally verifying gene functions is costly and slow, doing so for every gene is impossible. This gives rise to computational methods for automated function prediction, which can significantly reduce time and effort for manual verification. Automated function prediction is being routinely used as a powerful bioinformatics tool.

The definition of biological function must be clarified at the concrete context where it is used. For example, describing function of a protein as a catalytic enzyme is generally accepted, but not enough to specify the concrete catalytic activities, since enzymes can catalyze different chemical bonds and bind to different substrates. Thus, using proper function descriptor is essential for specific investigation purpose.

Generally, gene function is annotated in text form. In fact, the heaviest amount of function annotation is in text, which is stored in reference databases and literature. For text annotation, one may need a text mining tool to look for a function of interest. After all, text annotation requires human expert for curation. To enable automated function prediction, computer readable annotation is adopted. This form of annotation often specifies Gene Ontology terms and/or EC number as function's descriptors. However, while GO term hierarchy provides a general function description, for describing enzyme functions, EC number system should be used. Furthermore, for purpose of metabolic network reconstruction, the enzyme function must be known at substrate binding level (e.g. all 4 digits of EC numbers).

In this work, we focus on specific enzyme functions at substrate binding level. This section describes only those methods that can produce 4 digits of EC number prediction. There are several good reviews on automated function prediction [43, 44] and enzyme function prediction [45], which thoroughly describe each approach in details. Here, we give an overview on three main approaches for enzyme function prediction, namely homology-based, structure-based and network-based.

Homology-based methods

Sequence-based methods firmly rely on sequence similarity for homology transfer. The assumption is that, similarity of sequence infers similarity of function. The most widely used tools are BLAST and its variations. However, simple sequence similarity may mislead the homology transfer, due to several issues. Firstly, homologous enzymes may have low sequence similarity, since the enzyme functions may be "less conserved than anticipated" [46]. On the other hand, high sequence similarity may or may not be enough to transfer enzyme function at substrate binding level. Tian et al. [47] have shown that, to transfer enzyme function with 4 digit EC number, 60% sequence identity or higher is needed. Secondly, public annotation databases may contain erroneous annotation that can propagate throughout other databases. This is problematic since the data has been growing exponentially while expert labor for manual curation is expensive.

Liu et al. [48] combined different sequence-based kernels with evolution information to detect remote homology, which gives promising results. This multiple kernel-based approach can be extended with other sequence-based methods to further improve performance.

In fact, homology-based approach is the most original and effective for general purpose. Homology-based methods can sacrifice precision for coverage, by setting low cut-off threshold, and vice versus. The high coverage of this approach is dependent to the fact that much more homology data is available, in comparison with other genomics/post-genomics data.

Recently, a more favorable strategy for boosting accuracy is to consider small regions that are conserved, instead of the whole genome sequences. These conserved regions are distinct ***structural*** and ***functional domains*** of proteins. Millions of known protein sequences

(as in UniProt database [25]) are composed of only about 13 thousands of such domain families (as in CDD and Pfam databases). Shen et al. [49], Forslund et al. [50] and Messih et al. [51] used ***domain architecture*** as new insight for enhancing enzyme function prediction. However, these methods currently focus on classifying enzyme classes only at first level. Thus, more work is needed for predicting all 4 levels of enzyme classes.

Top-down homology-based approach

Top-down approach uses known family sequence profiles to identify protein domains/features, and map the results to biological functions. Pfam and CDD provide platforms for identifying such features, as well as a mapping between domain identifiers to EC numbers. A combination of EC2GO with Pfam2GO mappings (we called PF2GO2EC mappings), can translate domain identifiers into EC number annotation, via GO annotation. PROSITE [52] provides a library of profiles/patterns for recognizing enzyme functions. Methods using these domain features showed significant improvement on prediction accuracy. For example, standard PROSITE [52] patterns/profiles can predict EC numbers with an average precision of 92%.

Bottom-up homology-based approach

Bottom-up methods rebuild sequence profiles for all known enzymes, and then use these profiles to classify new enzymes. This approach is being more favored recently. Several bottom-up methods, such as PRIAM, EFICAz, ModEnzA, and CatFam [53] have been developed. A comparison between these methods is shown in Table 2.2.

Although these methods share a ***common classification protocol***, they differ in two important steps: *(1)* the strategy for properly clustering enzyme sequences into subgroups before building profiles, and *(2)* the threshold setting for enzyme profiles. PRIAM assumes that the shortest sequence in a collection (of the same EC number) is the unit of functional module, thus it uses that shortest sequence to search for similar regions in other sequences and then builds a PSSM for that functional module. This procedure repeats for the remaining set. Other methods, including EFICAz, CatFam and ModEnzA, simply cluster sequences based on the sequence similarity (BLAST e-value). For setting profile threshold, ModEnzA modifies HMM

profiles using negative sequences, and then performed cross-validation for optimizing the cut-off. PRIAM and CatFam simply use a predefined BLAST e-value for all profiles. For calculating the likelihood of a prediction, PRIAM assigns the hit's e-value to the prediction score. EFICAz and CatFam do not provide likelihood score for their final predictions.

Table 2.2 – A comparison on classification protocol of different methods

	EnzDP [54]	***PRIAM***	***EFICAz***	***ModEnzA***	***CatFam***
Similarity metric	Domain composition score by DEAS	Blast e-value	Blast e-value	Blast e-value	Blast e-value
Use of domain?	*Pfam domain*	*MKDOM functional module*	*No*	*No*	*No*
Clustering	*DEAS, rule*	*MKDOM, rule*	*MCL*	*MCL*	*MCL*
Profile type	HMM	PSSM	HMM	HMM	PSSM
Training source	Swiss-Prot	Swiss-Prot	Swiss-Prot	Swiss-Prot	Swiss-Prot
Profile threshold	Individual	Common (1e-30)	FDRs (functional discriminate residues)	-	Individual
Profile quality control	Optimized F1	By protocol	By protocol	By protocol	Optimized False Positive Rate
MSA	ClustalO	PSI-Blast	ClustalW	ClustalW	ClustalW
Profile builder	HMMER	PSI-Blast	HMMER	HMMER	Psi-Blast
Profile search	HMMER	PSI-Blast	HMMER	HMMER	Psi-Blast
Hit overlap check?	Yes	Yes	No	No	No
Classification rule	Yes	Yes	-	-	-
Active site check?	Yes	-	FDRs	-	-
Prediction score	Yes	Yes	No	-	No
Software avail.	Yes	Yes	Yes	-	Yes
Latest data avail.	Feb-2015	Mar-2014	Jan-2011	-	Oct-2008

Structure-based methods

While sequence and domain similarity may not specific enough to transfer functions, 3D structure of proteins of same functions is much more conserved. In other words, low sequence similarity proteins may still have significant structural similarity. Thus, using 3D structure to

predict function may achieve better accuracy. However, majority of known proteins have not been annotated for 3D structures, thus making this approach limited power, especially in coverage.

Given a sequence, one can scan the library of known 3D structure (PDB) to match structural similarity. After that, the function of the hit structures can be transferred to the query proteins. Multiple programs provide such searching, include FAST [55] , FATCAT [56], and VAST [57]. However, no method can give the completely accurate predictions, thus multiple methods should be used [58].

The catalytic site atlas database [59, 60] provides a resource for known active sites and catalytic sites of enzymes in their 3D structure. George et al [61] leverage on specific amino acid residues that matched with active sites to infer enzymatic function. However, the issue in all these methods is that, the set of known active sites is far from complete and currently covers a small fraction of known enzymes in Swiss-Prot [24]. Thus, the coverage of enzyme function predictable is much lower than that of using simple homology transfer.

Another database for enzyme structural annotation is PDBSum [35], which provides mappings between EC numbers and known 3D structures. Once the structure of a protein is predicted, it can be used to translate the corresponding EC number that the protein should be classified to. However, the accuracy of this approach strongly depends on performance of 3D structure prediction from sequence, which still needs significant improvement. On the other hand, as of June-2014, the current mappings contain only *one third* of all known EC numbers. This significantly limits the enzyme prediction coverage. Nevertheless, with increase of structural data, coverage of this method will be improved.

In short, protein structure can give a more specific and accurate indicator for determining biological functions. At the same time, the accuracy of structure-based methods depends on the ability of correctly predicting structure from sequences, which is still progressing. Furthermore, the coverage of these methods is currently limited by the number of known structures.

Network-based methods

Network-based methods [62, 63] use the association/interaction of genes in the same sub-network to infer gene function, based on the observation that genes closer in the gene association network have similar biological functions. Many types of association data have been used for inferring functional relationship, including: protein-protein interaction [64], gene clusters, genetic interactions, phylogenetics profiles [65] , gene clusters and operons [66, 67], and gene fusion [68].

The general idea behind network-based approach is the use of network-context in a functionally interact network of genes for inferring gene functions. The first step is to reconstruct such a gene network and locate the network neighborhood of the interested gene for function transferring. The gene of interest are assumed to have relevant functions to its direct/non-direct neighbors.

For reconstructing gene network, functional links between genes must be first identified. Several types of data can be used to infer functional links. The most high through put source of functional association data comes from *protein-protein interaction* (PPI) discovering [69]. The PPI data describes how proteins physically interact to co-function. Closely interact proteins may form protein complexes that work in a pathway to perform a specific function, thus having similar functions. The data can be identified by mass spectrometry techniques or in hybrid approaches. After that, a network of PPIs can be formed, re-scored and used for functional inferring.

The second type of functional association data is derived from *expression data*. Microarray expression technique allows measuring expression level of multiple genes at the same time. From experiment data in multiple conditions, expression profile of each gene can be generated. Using these profiles, pairs of genes that co-express or inhibit each other can be derived. These gene pairs with significantly correlated profiles will form a network of co-expressed genes. Finally, the network can further be used for inferring related functions of different genes [70, 71].

Many other types of functional association evidence can be used for network building, such as phylogenetics profiles [65], gene clusters and operons [66, 67], gene fusion [68], etc. These data provides plentiful resources for functional linkages establishment. Significantly, these types of data are increasingly generated and stored in public databases, such as BioGrid [72], MIPS [73], STRING [74], ProLinks, [75] and GeneMANIA [76].

After reconstructing a functional interaction gene network, network-based function prediction will be carried out. Most if not all methods base on the assumption that *the closer genes in network the more similar function between them* [62]. Thus, function of a gene can be inferred from its neighborhood in the network. The rule of Guilt-by-Association can be applied for set of direct neighbors of the gene of interest. However, function similarity may extent to indirect neighbors as well [77]. Thus, a neighborhood radius [78] should be considered for function transferring.

Significantly, in network-based approach, multiple data sources are combined to get a consensus measure of functional linkage. Chua et al. [77] proposed a simple yet effective and scalable FS-weight scheme to integrate heterogeneous resources, with different confidence weights. This method has shown that the combined network can improved significantly the performance of function prediction compared to single networks. Recently, GeneMANIA platform allows integrating multiple types of functional association for protein function prediction. These functional association data types can be retrieved from BioGrid and many other sources. Each type is then calibrated for a confidence score. After that, machine learning techniques are applied to learn the best fit combined network, as well as to predict the most functionally relevant genes for a given gene of interest.

Many other engines have also been developed for the purposed of network identification and integration. GENIES [79] utilizes supervised learning to infer unknown part of gene network from genome data. Olga et al. employed a Bayesian framework to combine heterogeneous function association data to predict protein function in yeast. More and more methods now focus on integrating data [80]. These methods increasingly give impact on new way to attack this challenge problem.

Interestingly, the combination of homology and network-based data has recently shown to be effective in predicting enzyme functions. Espadaler et al. [64] have attempted to combine sequence similarity with PPI data. The combination has led to 10% improvement in predicting 3 EC number digits over simple BLAST-based methods. Other types of combinations between network data and profile-based methods have not been tried. However, since using simple sequence similarity by BLAST can be easily improved with profile-based methods, the improvement of network-based data for profile-based method is not significant. While the network-based methods can work for non-homologous sequences, combine network data for profile-based may not work better than homology-based alone for homologous data, due to highly noise of functional association data. This also can be explained by the fact that homolog evidence is much stronger than functional association evidence. Thus, network-based methods should be used for global missing gene only, where homology-based is unable for the work.

In short, network-based methods exclusively use homology information, thus can be used as complement methods for homology-based. They also can work for non-homologous genes. However, the most significant limit of network-based methods is that, they cannot infer functions that more specific than functions of genes in network neighborhood, due to natural of the assumption behind. Furthermore, the functional data is often noisy, thus making prediction less precise. Another issue arises from the heterogeneity of functional association data sources, thus combining them is a challenge.

2.2.2 Network assembly problem

The problem of network assembly is described as following. Given a set of enzyme annotation of a genome, build a metabolic network that includes all these enzymes. The reconstructed network must be a *connected* network that *contains all given enzymes* and has *minimum size*. The first requirement is desired, since if the network is not connected, it obviously contains reaction gaps, which prevent analysis/simulation in later steps. If connectivity is not required, the set of initial enzyme reactions itself is a trivial solution to the problem. Due to this requirement, the network may contain reactions that do not have gene(s) annotated yet, which

are as known as metabolic gaps. For the second requirement, all known reactions are desired to be included so that no known activity is missing. The third requirement bases on the *parsimony assumption* that the minimum number of reactions happen to carry out all metabolic processes. Therefore, network size is expected to be minimized. This implies that, the reconstructed network must contain the *minimum number of metabolic gaps*, which shows a high quality reconstruction.

There are two approaches to network assembly problem, namely pathway-based and network-based. The first approach firmly relies on reference pathways database – such as KEGG [31] and BioCyc [41], while the second approach relies on reconstructed networks of closely related species, such as *Vongsangnak et al.* [3].

Pathway-based approach

This approach uses template pathways from KEGG to reconstruct network for a specific species. In fact, KEGG provides the most comprehensive set of reference biological pathways, with more than 480 pathways (as of May-2016). Each reference metabolic pathway contains all possible known reactions that can happen in that pathway. In other words, KEGG is a superset of all known pathways for all known species. In addition, it uses KGML format (KEGG meta-language), which is easily modifiable. Its pathway maps are manually drawn (in .png and .kgml format) and publicly available. Alternatively, BioCyc databases [41] contain metabolic pathways of many organisms, which were built by Pathway Tools software [81]. Many species specific pathway databases in BioCyc went through manual curation, however, the majority of databases is just the result of automated pathway prediction using Pathologic program [82]. Nevertheless, this provides an alternative to KEGG, with species oriented data.

The main challenges of pathway-based approach are to predict the presence of pathways and to define the pathway boundaries for specific genome of interest. Since not all reference pathways should be present in a specific species, the task here is to predict which pathways (of those 480+ KEGG reference pathways) the species has. Furthermore, not all reactions in a reference pathway are supposed to be included; therefore pathway-based methods have to

predict the pathway boundary (by deleting reactions that are not present), maintaining connectivity of the pathway.

Network-based approach

This approach uses the reconstructed network of relevant organisms to build an initial network first, then refine and enrich the network. For example, Vongsangnak et al. [3] used reconstructed networks of several fungi, namely *S. cerevisiae*, *A. niger*, and *A. nidulans* to merge into a template network for *A. oryzae*. According to this method, reactions that appeared in at least 2 referenced networks are included in the target network. After an initial network is reconstructed, the network is analyzed for connectivity. And the next step is to fill all the reaction gaps in the network to achieve connectivity for all the substrate – reaction – product relationship. In addition to building the network, network-based methods fill reaction gaps at this step. The purpose of filling reaction gaps is to add new reactions into the current network to connect all dead-ends, if any. Those new reactions are taken from reaction databases, and they may or may not have gene annotation available. All the methods in this approach try to minimize the number of new reactions, based on the parsimony assumption. This network reconstruction problem was proved to be an NP-hard problem [23, 83], due to the requirement of minimum reaction set.

Within the network-based approach, 2 streams of methods were developed, namely, *constraint-based* and *graph-theoretic* methods [83]. Constrained-based methods make use of experimental data (e.g., growth rate) for gap filling, such as SMILEY [84], GapFill [85], and GrowMatch [86]. The simulated data is compared with the experimental data to check for discrepancy. If there is any discrepancy, these methods will try to add new reactions into the model to reduce gaps, minimizing discrepancy. The gap filling problem is formulated as an optimization problem, and is then solved by optimization methods, such as integer linear programming. Differences in these methods are the problem formulation and rules of adding new reactions.

The computational method SCAR [87] follows the graph-theoretic framework. It uses reachability of the network itself to constrain the optimization formulation, without

experimental data. The network is reachable if all the reactions and metabolites are reachable. The network score is the sum of all its reaction scores, where reaction score is a sequence-similarity score (BLAST e-value) of the homologs. The optimization setting is to maximize the network score while maintaining its reachability.

2.2.3 Metabolic gap filling problem

Although there are reaction gaps and metabolic gaps, we focus on metabolic gaps, since reaction gaps are resolved in network reconstruction as a general task. Furthermore, filing metabolic gaps in metabolic networks is a complicated problem because the missing enzyme encoding genes have little supported evidence; otherwise they should be found already by reconstruction methods. The metabolic gaps are also called missing gene reactions, and the metabolic gap filling problem is also called missing gene problem, as described in [12].

Osterman and Overbeek [12] further classified metabolic gaps into 2 categories: local gaps and global gaps. A local metabolic gap is a metabolic gap in the reconstructed network of a species. However, it is not a gap in the reconstructed network of other species. This means that, the enzyme genes were identified in other species, but missed in the species of interest. On the other hand, a global metabolic gap is a gap that appears in all known species. In other words, the globally missing gene has not been found in any known genome yet.

Metabolic gap filling problem is following: Given a metabolic network of a target organism and set of missing gene reactions, find in the target genome the genes that encode the corresponding enzymes. Note that, this problem is a duo problem with the function prediction problem (see Chapter 3 for more details).

General frame work for metabolic gap filling

A typical procedure for filling metabolic gaps consists of 3 steps [12], namely identifying a list of candidate genes, evaluating the candidates and finally, experimentally verifying. The first two steps can be approached by computational methods. However, the last step needs wet-lab in vivo experiments, which is beyond the capacity of computation. Despite that, good result of in silico prediction would greatly reduce time and effort for the in vivo verification.

Generally, there are two types of clues for identifying candidates for missing gene reactions. First, the missing gene may have significant sequence similarity (homology) with the known ones in other networks and/or reference databases that have the same functions. Second, the missing gene may have a functional association with other known genes in the same network (context). At the same time as candidates are identified, they can be evaluated by its sequence similarity score (homology) or functional association score (context).

Table 2.3 – Method for finding missing genes in metabolic networks

Methods and reference	**Type of evidences**			**Organisms under study**
	Homology	Network structure	Association evidences	
GFAOP, *Vongsangnak et al., 2008*	√		None	*Aspergillus Oryzae*
Reed et al., 2003	√		None	*E. Coli*
MEP, *Kharchenko et al., 2004*		√	Gene expression	*Yeast*
ADOMETA, *Chen & Vitkup, 2006*		√	Phylogenetic profiles	*Yeast, E. Coli*
ADOMETA, *Kharchenko et al., 2006;*		√	Phylogenetics, expression profiles, gene cluster, gene fusion	*Yeast, E. Coli, B. Subtilis*
Yamanishi et al., 2007		√	Chromosomal proximity, phylogenetics profiles, (and chemical information)	*P. Aeruginosa*
Pathway Hole Filler, *Green& Karp, 2004*	√	√	Operon, gene cluster	*PGDBs, CauloCyc...*

Based on those clues, three categories of methods have been developed to find missing genes in metabolic network, namely homology-based methods (GFAOP [3], Reed et al. [88]), context-based methods (Yamanishi et al. [89], ADOMETA [90-92], Yamada et al. [93]) and hybrid methods (PHFiller [82]). Homology-based methods use sequence similarity for both identification and evaluation of candidates, while context-based methods use gene association evidences for identification candidates, and then either ranking candidates by a chosen cost function or calculating the probability that the candidates have the desired function. A hybrid method, as a combination of those two, makes use of both methods. Table 2.3 lists current methods for metabolic gap filling.

Current methods for metabolic gap filling

This section give a discussion on several state-of-the-art gap filling methods, namely: Gap Filler for *A. oryzae* Pathway (GFAOP), ADOMETA, Yamanishi et al. and Pathway Hole Filler (PHFiller). Summary of advantages and disadvantages are shown in Table 2.4.

GFAOP

GFAOP, which was developed by Vongsangnak et al. [3], can be considered as a representative of gap filling methods that use homology reference. The reference databases used are Pfam, COG, and NR. The COG (Clusters of Orthologous Groups of proteins) database maintains a system for classification of genes based on orthologous relationship, hence providing a comparative genomics framework for genome annotation and evolutionary studies. The Pfam (Protein family) database collects all protein families that are conserved across several species. It also integrates bioinformatics tools for generating sequence profiles for each family such as HMMER. Hence, it is very convenient using Pfam to identify enzyme sequences for given functions/family descriptions. On the other hand, NR (Non-Redundant protein sequence) database is a huge collection of all non-redundant protein sequences from other comprehensive databases (such as Swiss-Prot, PIR, PDB, GenBank, EMBL, etc.). These databases are useful in genome functional annotation and manual validation.

In the first step, GFAOP takes in an identifier of the protein family that is known to have required metabolic function of the gap. After that, it searches over Pfam, COG databases for the Hidden Markov Models profiles for the family. In the second step, the profiles are used to search against the target genome for candidates using PSI-BLAST. Finally, top candidates are validated using NR database.

This homology-based method is simple yet effective, especially for reconstruction from scratch. It filled 210 over 278 metabolic gaps in initially reconstructed network of *A. oryzae*, which leads to first version of *A. oryzae* network (*iWV1314* [3]). However, a serious drawback of GFAOP is that, it requires the enzyme family must be known as input. The first step was done with help of expert curator, who knows what to look for. It may fail in the very first step if the family is not well defined, or not known yet.

ADOMETA

ADOMETA, which stands for "Adoption of Orphan Metabolic Activities", was developed from combination of three works (Kharchenko et al., 2004 [90], Kharchenko et al., 2006 [92], and Chen and Vitkup, 2006 [91]). ADOMETA rebuilds the metabolic gene network from the reconstructed metabolic network. Two genes are connected if their corresponding reactions share at least one common metabolite. A metabolic gap corresponds to an unknown gene in the gene network. Neighborhood distance up to 3 is considered when dealing with gaps.

Chen and Vitkup [91] used phylogenetic profiles to score the similarity of gene neighborhood. The authors found that, the closer genes in the metabolic gene network, the more similar evolution history they have. Thus, candidate genes for a gap must have similar phylogenetic profiles with the gap's neighbors. Leveraging on this hypothesis, they propose several cost functions to rank gap's candidates. Furthermore, optimization techniques (such as simplex and simulated annealing) were used to tune cost function parameters, training on the set of known metabolic genes. The best function was then used for filling gaps.

Together with phylogenetic profiles, Kharchenko et al. [92] used other type of functional association data, such as gene expression correlation, gene fusion, protein-protein interaction, gene cluster, etc. Combining these association data significantly improved the prediction performance. The most advantage is the use of association data instead of homology, thus it can fill global gaps. Furthermore, this method makes use of network structure. However, association evidence is not as strong as homology evidence, thus the method has low precision. For example, for yeast *iFF708* network, ADOMETA correctly predicted only 60% of known enzymes with a cut-off of top 50 candidates.

Table 2.4 – Advantages and disadvantages of current gap filling methods

	Advantages	**Disadvantages**
GFAOP	Simple yet effective, make use of other available tools; Works well for newly reconstructed networks, such as metabolic network of A. Oryzae (210 out of 278 missing genes are predicted);	Can find only local missing genes; Gene family must be identified first; Does not make use of network structure, considers missing gene reactions as individuals; Candidates are not clearly evaluated;

ADOMETA	Combine multiple association evidences for both identification and evaluation of candidates; Consider missing gene reactions in context with neighborhood relationship; Could find global missing genes;	Large amount of "expensive" data should be analyzed, such as building phylogenetic profiles, gene expression profiles, Association evidence is not as strong as homology evidence Require that networks are initially built;
Yamanishi et al., 2007	Use genomic context and functional association evidences; Make use of network structure; Guide select candidates based on chemical information described in EC number.	Gene location information may not be useful for eukaryotes; Require building gene interaction network; 3 digits of EC number are not enough to confirm specific metabolic function.
PHFiller	Use both homology and association evidences; Can be generalized for using multiple sources of evidences.	Can find only local missing genes; Function of the missing gene must be well defined/characterized in reference species; Genomic context (operon) evidences may not applied for eukaryotes;

Yamanishi et al.

This method (described in [89]) uses KEGG databases as reference. This database contains a large collection of genomic information, from orthologous gene clusters to reference metabolic pathways. Genomic context and phylogenetic profiles are used as clue for finding missing genes. It applies machine learning technique with supervised approach to infer the gene interaction network and at the same time, evaluate the gene correlation. It not only proposes candidates but also guides selecting the right one by comparing first 3 digits of their EC numbers.

An advantage of this method is that it uses machine learning technique for inferring gene network, and then compares with KEGG reference pathway to identify missing gene candidates. However, the same as ADOMETA, this method requires the network to be at least partially reconstructed, thus cannot do de novo reconstruction. Significantly, the genomics context may not strong for eukaryotes, thus making them limited power.

Pathway Hole Filler

Developed by Green & Karp [82], PHFiller is a hybrid method, which combine homology evidence with genome context to predict gap candidates. Similar to the homology-

based methods, this method uses homology information as the main source to identify candidate genes. In addition, it takes into account additional information from genomic (operon) and functional context, as well as pathway context to evaluate the fitness of a candidate. Simple Bayesian classifiers were introduced to calculate the posterior belief of the prediction. The use of the Bayesian classifier is to reject or accept a candidate gene.

The method achieved high precision and has advantages in comparison to GFAOP. First, it not only uses homology information, but also genomic and functional context information. Hence, the proposed candidates would be more reliable. Second, it can be generalized to combine multiple fitness evidences (e.g. other non-homology information) into a consensus measure by the Bayesian classifiers. Thus, it is more flexible with availability of data.

However, since genome context is not strong for eukaryotes, this method is not well applied on them. Significantly, PHFiller heavily relies on homology, which makes it incapable of finding novel genes. The method performs best only for initial reconstruction. Furthermore, calculation of Bayesian classifiers using reference may be database-bias, thus its result depends greatly on how closely related the target genome with reference databases is.

Chapter 3

MeGaFiller: retrofitting function predictors for filling metabolic gaps

To improve network quality, one significant procedure is to fill the gaps after the network has been reconstructed. The problem of metabolic gap filling is well known [11, 12] and is often referred to as missing gene problem. This chapter describes our approach to tackle this metabolic gap filling problem.

Notably, filling metabolic gaps in reconstructed network is difficult. This is because these gaps have no or little evidence to find, otherwise they have been found in previous steps of network reconstruction already. Significantly, these gaps may be missed by previous gap filling methods that were performed during network reconstruction. Previous homology-based gap filling methods are generally based on identifying a protein family in related organisms and then using this family to help finding the target gene in a given genome. However, these methods fail when the protein family is not well-defined. There are therefore still many gaps in current metabolic networks. For examples, in the reconstructed metabolic networks of yeast *Saccharomyces cerevisiae [2]*, filamentous fungi *Aspergillus oryzae [3]*, *Aspergillus nidulans [5]*, and *Aspergillus niger [6]*, and bacterium *Streptomyces coelicolor [94]*, from 6% to 19% of the unique biochemical reactions are metabolic gaps. Here, we attempt to fill these gaps via an indirect approach by retrofitting protein function predictors and post-processing their results to identify the candidate genes.

As a result, we developed a novel method for metabolic gap filling, called MeGaFiller that uses an ensemble of multiple retrofitted state-of-the-art protein function predictors. The ensemble scheme was adopted to boost the prediction performance. MeGaFiller can propose the candidate genes for 35% of the metabolic gaps in different metabolic networks (i.e. yeast, three filamentous fungi and bacterium). MeGaFiller can predict novel candidate up to hundreds genes for earlier annotated functions in the metabolic networks. MeGaFiller can also provide novel candidate genes for novel putative reactions throughout the metabolic networks.

MeGaFiller method demonstrates our first effort for filling metabolic gaps in the metabolic networks by retrofitted protein function predictors. It serves as a bioinformatics tool assisting for improved annotation through metabolic network reconstruction at a genome-scale.

Outline of this chapter is following. First we describe the issues of current direct gap filling methods and overview of our approach to overcome them. Next we give details of our method, how we retrofit function predictors into gap fillers and integrate them into MeGaFiller. After that, experimental results as well as gap filling results will be presented. Finally discussion and conclusions will be given.

3.1 Background

3.1.1 Current metabolic gap filling methods

Current direct methods for filling local metabolic gaps (i.e. genes that are un-annotated in the target organism, but have been found in other related organisms) are based on the profile of the protein family, such as GFAOP [3] and Reed et al. [88]. In these methods, the first step is to identify the specific protein family for the metabolic gap. The next step retrieves and/or builds a family profile of this protein from reference databases. Then this profile is used to search against the target organism (i.e. genome sequence) to detect candidate gene. Finally, candidate gene (if any) is manually validated by querying with annotation databases. These gap filling methods have been successfully used to fill some of the gaps in the metabolic networks of *Escherichia coli* [88] and *A. oryzae*. Despite that, many gaps still remain in these networks.

The success (and failure) of these direct gap filling methods is highly dependent on the first step of identification of the protein family for the metabolic gaps. This step requires an expert to precisely interpret the proper family that performs the intended enzymatic function (described by an EC number) of the metabolic gap, specific up to substrate binding. However, this first step is challenging for two reasons. *Firstly*, the protein family is not well-defined. For example, in the Swiss-Prot database, there are 251 annotated genes encoding for the EC:2.1.1.64. Of these, 242 genes have no known specific protein families which are available.

Secondly, even if the protein family is found, it may not be particular enough to be helpful for finding the candidate gene with a specific function. Based on Swiss-Prot database, for example, there are a total of 673 annotated genes encoding for the EC:2.7.7.3 and all of them carry the *Cytidylyltransferase* (PF01467) domain. Meanwhile, there are the other 527 annotated genes encoding for the same domain, but show different functions. This indicates that the domain is too general and hence the generated protein family profile will not be specific enough to find good candidate gene for the EC:2.7.7.3. We noticed that a current *A. oryzae* metabolic network, 52 gaps (out of 61 gaps) do not have a specific protein family interpretation. Hence, these direct gap filling methods have failed in the beginning step as shown in example case of the *A. oryzae* metabolic network (*iWV1314*).

3.1.2 Overview of our approach

In this work, we aimed to develop MeGaFiller, an ensemble of indirect approach to overcome the difficulties of existing direct methods in cases of poorly characterized protein family. Our indirect approach leverages the following duality between gap filling and protein function prediction. In gap filling, we determine protein function (*f*), and we desire to search for candidate gene (*p*) in the genome sequence of the target organism with the protein function f. In protein function prediction, we determine a candidate gene p, and we desire to predict its protein functions *f*. Thus, gap filling and protein function prediction are dual problems of one another.

In theory, therefore, if we have a candidate gene p in the target organism (genome *G*) that performs a protein function *f*, then the "perfect" gap filler is able to find gene *p* in *G* given the protein function *f*, and the "perfect" protein function predictor is able to predict protein function *f* given the gene *p* in *G*. In reality, however both procedures are far from the "perfect". Current direct gap filling methods are not able to find the genes encoding for some protein functions, as evident from the gaps in current metabolic networks. Additionally, current protein function predictor may miss some protein function *f* due to very low scores from the prediction.

We propose to use the dual approach using protein function prediction methods to help find candidate gene p that have the predicted protein function *f*. We initially predict the functions of all the proteins in a target organism (genome *G*) using a protein function predictor. Then, we keep only the genes with predicted protein functions that are matched to those of the metabolic gaps. Once completed, the list of candidate genes is generated for the metabolic gaps.

There are additional reasons to pursue this dual approach. Firstly, there has been tremendous progress in the state-of-the-art protein function prediction methods and they have vastly improved recall and precision rates. Many enhancements have been developed for BLAST-based protein function predictors and these included Gotcha [95], PFP [96, 97], and Blast2GO [98, 99]. There have also been enhancements to protein function predictors that used Hidden Markov Model (HMM) profiles with improved prediction accuracy and these included ModEnzA [39] and EFICAz [37, 38]. Hence, it is timely to leverage on these state-of-the-art protein function predictors to help find candidate genes for the "difficult-to-fill" metabolic gaps. Another crucial reason is that our dual approach does not require knowledge of the specific protein family and hence gets around the inherent challenge of identifying the proper family at the beginning. Thirdly, even in case where the protein family is not well-defined, there may still be some of individual protein list in the annotation databases that can infer function. Several protein function predictors can leverage on the protein list (via sequence similarity for BLAST-based methods or profile similarity for HMM based methods) to help in their function prediction. In this way, the protein list may help, indirectly, to predict the given metabolic function for achieving the candidate genes, without having the certain protein family.

To successfully use this dual approach, we identified good protein function predictors that were suitable for retrofitting for the purpose of filling metabolic gaps. We focused on protein function predictors that gave more predicted functions, even those with low scores. To deal with a large pool of predictions, we needed a retrofitting procedure to carefully filter and find the "difficult-to-fill" candidate genes for the metabolic gaps (details are given in the next section). Based on this, we retrofitted gap fillers based on the different state-of-the-art function predictors. After evaluating their individual effectiveness, we found that no one method dominates the rest and each of them has different strengths and weaknesses. General protein function predictors tend to give many more predicted functions (per protein) and thus, achieve higher coverage, but lower accuracy. In contrast, enzyme-specific function predictors give fewer predicted functions (per protein) and gain higher accuracy, but lower coverage.

To leverage on their different relative strengths, we developed novel MeGaFiller (Metabolic Gap Filler) method that uses a weighted ensemble of the individual retrofitted protein function predictors. We then optimized the relative weights of the individual protein function predictors within MeGaFiller. This optimization of MeGaFiller was then performed separately in each of five species, as our performance analysis showed that the optimal parameter setting is species-dependent. To the end, the optimized MeGaFiller was used to fill the gaps in five different metabolic networks. MeGaFiller showed effective in filling metabolic gaps remaining in these networks. It was also able to predict more candidate genes for existing reactions and novel putative reactions in these metabolic networks.

3.2 MeGaFiller method

Our proposed method is called MeGaFiller, which carefully combined the prediction results of several individual gap fillers based on retrofitted protein function predictors. In the following, we first explain how these individual gap fillers were designed, and then discuss how they were retrofitted into MeGaFiller.

In the dual approach for gap filling, we used protein function predictors to indirectly find candidate genes that have the predicted protein functions of the metabolic gaps. For a given

metabolic network of a target genome *G*, and any chosen protein function predictor (*FP*), the general procedure is as follows: First, we used *FP* to predict the functions of all the proteins in a target genome. This produces, for each gene *p* in the target genome *G*, a list of predicted protein functions (given in EC numbers or GO terms). Usually, these predicted protein functions were ranked based on a variety of scores (e.g. confidence, significance) depending on the predictor used. Metabolic gaps were given by their EC numbers. Hence, the next step was to use EC2GO mapping [21] to map any predicted GO terms into EC numbers. Next, we matched these predicted protein functions with the gaps in the metabolic network. For each metabolic gap with EC number, we collected all the genes for which the protein function *f* that was predicted by *FP* and formed the list of candidate genes for the metabolic gaps.

We pointed out that the ranking/scores of candidate gene *p* for a given metabolic gap were produced by considering candidate gene *p* individually, and the ranking/scores were given relative to the other predicted protein functions for that candidate gene *p*. Hence, it does not make sense to directly compare the ranking/scores of genes in the candidate list. Hence, we may need to do some retrofitting by post-processing of the candidate list (for example, re-ranking by their confidence/significance scores) to produce the list of missing gene candidates.

With the recent advances and proliferation of protein function predictors, we selected suitable ones. For our purposes, we focused on state-of-the-art protein function predictors that are also relatively easy to retrofit for metabolic gap filling. We thus selected two general protein function predictors, PFP and Blast2GO that can predict general protein functions. We also selected an enzyme-specific function predictor, EFICAz. We ran all the three protein function predictors using their default settings.

3.2.1 Retrofitting procedures

In this section, we describe how we retrofitted function predictors for gap filling and called them as PFP-GF, B2G-GF, and EFICAz-GF, respectively.

PFP-GF: Retrofitted PFP for Gap Filling

Among the general protein function predictors tested, PFP gave the most predicted functions with the best coverage (but low accuracy). Given an input as protein sequences, PFP uses PSI-BLAST on the UniProt database to predict a list of GO terms, each of which comes with several scores (i.e. rank, raw score, P-value, and three different confidence scores). By default, PFP sorts GO terms by the "4-edge confidence score". This list contains many predicted protein functions (some may contain up to 500 GO terms), including many predicted protein functions with very low scores.

We retrofitted the protein function predictor PFP for gap filling as shown in Figure 3.1. After PFP was done, the output GO terms were mapped into EC numbers using EC2GO. Then, for each EC number, we re-ranked the list of candidate genes based on the following criteria (in priority order): confidence score, raw score, and P-value. A top-rank cut-off to filter the candidate list was chosen to maximize the F_2 score. The best F_2 score for PFP-GF was obtained when setting this cut-off to keep only the top 5 candidates.

B2G-GF: Retrofitted Blast2GO for Gap Filling

The other general protein function predictor, Blast2GO, is very simple to retrofit. Blast2GO produces EC numbers for their predicted functions. It produces relatively few predicted functions per proteins and they tend to have high accuracy. Thus, the retrofitting procedure consists of just matching predicted functions per proteins with the given set of EC numbers, and consolidating all the candidate genes for each of the given EC numbers.

EFICAz-GF: Retrofitted EFICAz for Gap Filling

EFICAz is an enzyme-specific function predictor. It predicts only enzyme-specific functions. It gives relatively few predicted functions and these have high accuracy. The predicted protein functions are also given by their EC numbers. Thus, like B2G-GF, retrofitting consists of just matching predicted functions per proteins with the given set of EC numbers, and consolidating all the candidate genes for each of the given EC numbers.

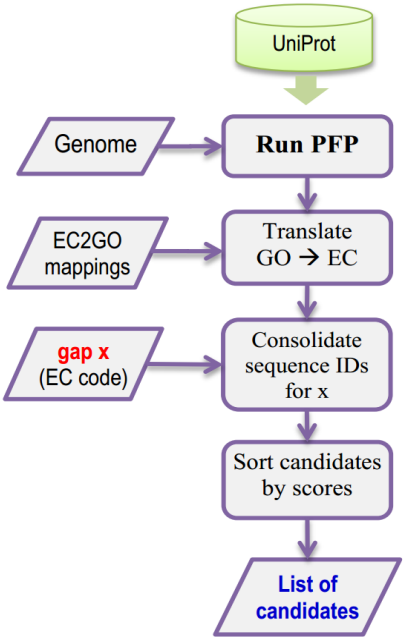

Figure 3.1 – Retrofitting procedure for gap filler PFP-GF

The original PFP takes a genome sequence as an input and generates a list of predicted GO terms for each sequence. PFP-GF takes the PFP's output, the EC2GO mapping, and a metabolic gap described by an EC number. Then, PFP-GF generates the sorted list of candidate genes for the gaps.

3.2.2 Our proposed method: MeGaFiller

To achieve better prediction results and higher confidence, our proposed method, MeGaFiller, considers an ensemble of these three individual gap fillers. For this integration, we need to handle the fact that the three individual gap fillers produce different types of prediction scores. Specifically, PFP-GF gives several scores, and, after our performance analysis, the confidence score was chosen as the score for PFP-GF. In contrast, B2G-GF ranking relies on the BLAST hit score, thus the best bit score of hits was chosen as the score of candidates for B2G-GF. Finally, EFICAz-GF does not produce any prediction score, and so all its predictions were equally weighted (i.e. all have score of 1.0).

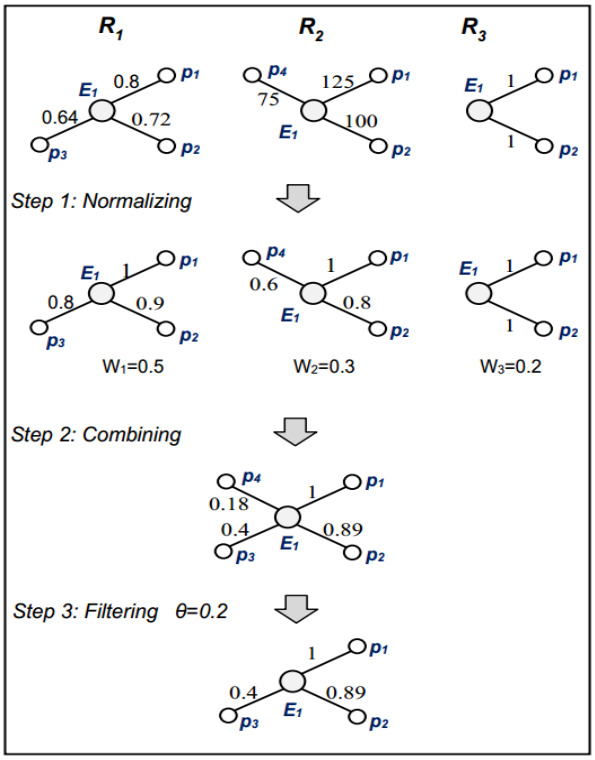

Figure 3.2 – Ensemble scheme for MeGaFiller

A big circle denotes an EC number, while a small circle presents a candidate gene. Scores of predictions made by each predictor were normalized. Then, the predictions were combined by the weight w_i of each predictor. Finally, MeGaFiller applied a threshold θ to filter the final prediction list.

To rationally combine these gap fillers, a generic weighted ensemble scheme (similar to [77]) was adopted, as shown in Figure 3.2. Firstly, for each EC number (e), the scores for each candidate gene p for e were normalized using as follows:

$$S_N^i(e,p) = \frac{S^i(e,p)}{max_k\, S^i(e,p_k)}$$

where $S_N^i(e,p)$ is the normalized score and $S^i(e,p)$ is the predicted score of pair (e, p) produced by gap filler (i). Secondly, we assigned weight (w_i) to each gap filler i as a measure of its prediction significance. These were also normalized, i.e.:

$$\sum_i w_i = 1$$

The combined score, $C(e, p)$, for a pair of a candidate gene p and an EC number e, was given by a weighted summation over all individual gap filler, as follows:

$$C(e,p) = \sum_i w_i \, S_N^i \, (e,p)$$

To the end, the candidate gene list was filtered, keeping only candidates with the combined scores not below a given threshold (θ).

3.2.3 Data sources and performance metrics

In this research, we focused on using MeGaFiller to fill metabolic gaps in the reconstructed metabolic network of *A. oryzae iWV1314*. However, we have also run MeGafiller on the other four networks, those for *S. cerevisiae iIN800*, *A. niger iMA871*, *A. nidulans iHD666*, as well as *S. coelicolor iIB711*. These metabolic networks were obtained from BioMet Toolbox [100] website (http://biomet-toolbox.org/)

Each of these metabolic networks contains two datasets: the known dataset (N_K) and the metabolic gap dataset (N_G). N_K is a list of known pairs of metabolic reaction associated between EC numbers and gene. The N_K was used to tune the parameters of MeGaFiller and its component gap fillers. N_G is a list of metabolic gaps associated between EC numbers and gap reactions. From each metabolic network of the five species, we extracted N_K and N_G and the statistics are shown in Table 3.1 and Table 3.2.

Table 3.1 – Metabolic networks used in this study

Network	Species	Strain	Release	Genome Source
iIN800	*Saccharomyces cerevisiae*	S288c	2008	SGD[1]
iWV1314	*Aspergillus oryzae*	RIB40	2008	DOGAN[2]
iHD666	*Aspergillus nidulans*	FGSC A4	2008	AspGD[3]
iMA871	*Aspergillus niger*	ATCC1015	2008	AspGD[3]
iIB711	*Streptomyces coelicolor*	A3(2)	2005	*S. coelicolor*[4]

[1] http://www.yeastgenome.org/
[2] http://www.bio.nite.go.jp/dogan/top
[3] http://www.aspgd.org/
[4] http://www.sanger.ac.uk/resources/downloads/bacteria/streptomyces-coelicolor.

Table 3.2 – Metabolic network contents

Network characteristics	iIN800	iWV1314	iHD666	iMA871	iIB711
Number of gene-EC number pairs	573	1,329	847	818	694
Number of unique EC numbers	481	711	455	482	407
Number of metabolic rxns w/o genes	83	65	29	112	72
Number of unique EC numbers without genes (number of gaps)	***52***	***61***	***28***	***90***	***68***
Percentage of metabolic gaps	11%	9%	6%	19%	17%

Note: This table shows detailed content of the five metabolic networks. Unique EC numbers are accounted for the set of EC numbers appeared in the network. Number of metabolic gaps is the number of unique metabolic reactions that have no genes annotated. Percentage of gaps is calculated as ratio of gaps over unique EC numbers

We retrieved genome sequences of these five species from reference databases (*S. cerevisiae* S288c – SGD (http://www.yeastgenome.org/) [101], *A. oryzae* RIB40 – DOGAN (http://www.bio.nite.go.jp/dogan/top) [42], *A. niger* ATCC1015 and *A. nidulans* FGSC A4 – AspGD (http://www.aspgd.org/) [102] and *S. coelicolor* (http://www.sanger.ac.uk/resources/downloads/bacteria/streptomyces-coelicolor.html) [103]. We used these genomes to extract the list of protein sequences to serve as input to MeGaFiller (and to the individual metabolic gap fillers).

To tune the parameters of MeGaFiller and its component gap fillers, we ran them on the known datasets N_K of the five species and measured the following performance metrics: precision, recall, and F_2 score, calculated as follows: Suppose that a gap filler method proposes a candidate gene *p* for an input EC number *e*. Then the predicted pair (*e*, *p*) is said to be a true positive if (*e*, *p*) is in the N_K; otherwise it is called as a false positive. A pair (*e*, *p*) is called a false negative if it is in the N_K, but is not predicted by the gap filler. Let *TP*, *FP* and *FN* denote the number of true positive, false positive and false negative, respectively. Then the performance metrics are calculated as the following:

$$Recall = TP / (TP + FN)$$

$$Precision = TP / (TP + FP)$$

$$F_2 = 5 * Precision * Recall / (4 * Precision + Recall)$$

We used F_2 score to put more weight (double the importance) on the prediction coverage (recall), which is more important than the prediction precision for the purpose of finding missing gene candidates.

The aim of MeGaFiller is to find candidate genes to fill gaps that existing direct gap filling methods have already failed. So, MeGaFiller is biased towards higher true positives and finding new candidate genes, at the expense of an increase in the number of false positives. In other words, it is more important to be able to find the candidate genes while incorrect predictions may be ruled out later by manual curation with the help of additional independent data sources.

3.2.4 Parameter tuning for MeGaFiller

MeGaFiller has several parameters, the weights w_i for the component gap fillers and the threshold (θ). We tuned parameters using the known datasets as follows: Consider a species with genome G and reconstructed metabolic network N as $N_K(G)$. Let $R(G)$ is the list of metabolic reactions (given by their EC numbers) in $N_K(G)$. For any given values of the parameters w_i and θ, we ran MeGaFiller using the genome G and the $R(G)$ to predict pair (e, p), namely to predict gene p in G for the EC number e in $R(G)$. Let $P(w_i, \theta, G)$ denotes this list of predicted pair obtained by MeGaFiller with parameters w_i and θ. We then matched the predicted pair with $N_K(G)$ to compute $F_2(P(w_i, \theta, G) \mid N_K(G))$, the F_2 score for this parameter setting. These parameters are tuned by optimizing the F_2 score:

$$(w_i, \theta)_G = argmax\ (F_2(P(w_i, \theta, G) \mid N_K(G)))$$

All parameters were searched in the range (0,1) with step-size of 0.01.

3.2.5 Manual curation of candidate genes for metabolic gaps

For new predictions made by MeGaFiller (and others methods) to fill the metabolic gaps in various networks, there is no ground truths. To evaluate, hence the reliability of these new predictions by MeGaFiller, we manually curated some of them to provide independent supporting evidences. For each predicted pair (e, p), we used the protein sequences of the candidate gene p to search against several annotation databases, such as NR

(http://www.ncbi.nlm.nih.gov/refseq/), UniProt, Pfam, CDD and KEGG databases to look for significant hits with annotation of the function *e*. We also looked at the updated function annotation of the candidate *p* if such annotation exists in genome databases of the species. All the relevant supporting information for the candidate gene *p* was collated for further manual curation.

3.3 Results

We now present our extensive results which are organized as follows: We first show an evaluation of individual retrofitted gap fillers using the known datasets from the five species studied. We show how these results supported our use of an ensemble scheme in MeGaFiller. Next, we provide results on parameter tuning of MeGaFiller. We show that MeGaFiller (with default optimal parameters) outperformed the individual component gap fillers with the highest recall and F_2 score.

We then describe the main result of this study, namely using MeGaFiller for gap filling and the ability of MeGaFiller to fill critical gaps in the *A. oryzae* network. Besides, we then show comparison of our method, MeGaFiller with two other existing methods: GFAOP, a homology-based method and ADOMETA, a context-based method. Finally, we discuss how to use MeGaFiller to predict novel candidate genes for existing reactions and/or predict novel putative reactions in the metabolic network for a given species.

3.3.1 Evaluation of the individual retrofitted gap fillers

To evaluate the relative performance of the retrofitted gap fillers (PFP-GF, B2G-GF, EFICAz-GF), we ran each of them on the same known datasets of the different species. A completely uniform comparison was not possible since each method relies on its own (different) reference data that were retrieved at different timestamps. To do a fairer comparison, we restricted the comparison to only those reactions that are found in all three methods. We also excluded those EC numbers that were too general and focused on those that were specified in all 4 digits. We

noted that, across the 5 genomes, from 1.0% to 8.3% of the pairs, were excluded by this process (and overall of only 4.7%).

After this pre-processing, the known dataset for metabolic network *iIN800* contains 573 known gene-EC number pairs, while the known dataset for network *iWV1314* contains 1,329 known pairs. (The numbers of gene-EC number pairs in known datasets for *iMA871*, *iHD666* and *iIB711* are 818, 847 and 694 pairs, respectively).

Figure 3.3 shows the prediction results of PFP-GF, B2G-GF, and EFICAz-GF on the known datasets for *iIN800* and *iWV1314* datasets. For the well-studied yeast *iIN800* dataset, PFP-GF gave the largest number of true positive predictions, 520 out of 573. This was followed by EFICAz-GF with 445 true positive predictions while B2G-GF gave 402 true positives. However, PFP-GF also predicted more false positives (1,438) compared to B2G-GF (164) and EFICAz-GF (119).

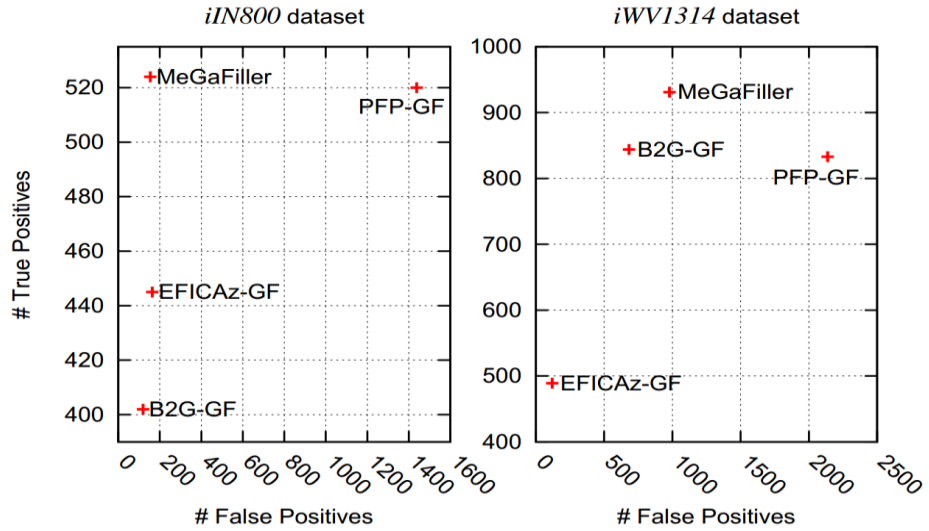

Figure 3.3 – Performance evaluation of retrofitted gap fillers

The dot plots show the number of true positives and false positives on iIN800 (*S. cerevisiae*) dataset (left), and iWV1314 (*A. oryzae*) dataset (right) by component gap fillers and the MeGaFiller.

For the relatively less well-studied filamentous fungus *iWV1314* dataset, PFP-GF and B2G-GF gave many more true positive results (833 and 844 out of 1,329) than EFICAz-GF (445). PFP-GF also had the highest false positives (2137), while the enzyme-specific EFICAz-

GF had the lowest (119), and B2G-GF was in between (682), but closer to EFICAz-GF. The results for the other three genomes (also less well-studied and annotated) were similar to that for *iWV1314* and are not shown in Figure 3.3.

We combined the results for all five genomes and observed that the retrofitted general protein function predictors gave higher average recall (PFP-GF: 67.5%, B2G-GF: 57.2%) compared to retrofitted enzyme-specific function predictors (EFICAz-GF: 46.4%). EFICAz-GF has higher precision (77.5%) than B2G-GF (62.4%) and PFP-GF (27.4%).

This can be explained by the fact that, enzyme-specific function predictors like EFICAz focuses on accurately classifying only enzymatic functions. Hence, they tend to make fewer predictions that are more accurate (higher precision) and they may lose in prediction coverage (lower recall), especially on datasets of less well-studied species. On the other hand, PFP, being a general protein function predictor gave more predictions than either EFICAz or Blast2GO, but suffered from lower precision.

3.3.2 Evidence to support an ensemble approach

We compared the actual predictions made by PFP-GF, B2G-GF, and EFICAz-GF for the *iIN800* and *iWV1314* datasets. The Venn diagram is shown in Figure 3.4. For each species, we observed that the number of true positive predictions in the 3-way common intersection is quite small, and each of gap filler produced its own unique true positive predictions. We also observed that PFP-GF gave the most unique true positive predictions, but with a high false positive rate. Thus, combining the prediction results of PFP-GF with that of B2G-GF or EFICAz-GF will increase the *precision.*

More generally, this suggests that we need to leverage on the results of all the gap fillers. Our method, MeGaFiller used a weighted ensemble scheme to combine the results of all three gap fillers to leverage on the high recall of PFP-GF and the high precision of the B2G-GF and EFICAz-GF.

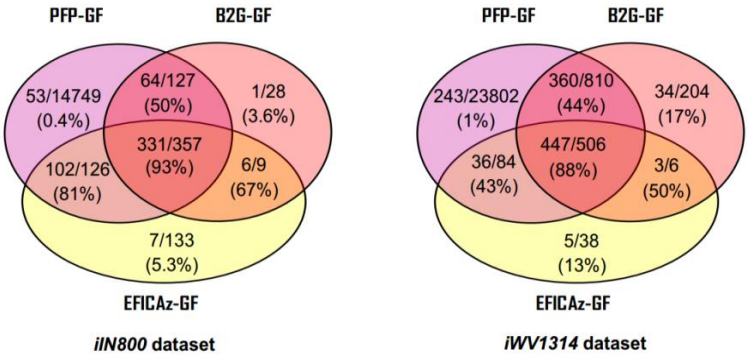

Figure 3.4 – Intersection of predictions made by different gap fillers

Common predictions made by different methods and the corresponding correctly predicted portion for iIN800 (left) and iWV1314 (right) datasets. There are 357 predictions made by all gap fillers on iIN800 dataset, in which 331 predictions are correct (93%). None of methods can cover all possible correct predictions.

3.3.3 Parameter tuning for MeGaFiller

To tune MeGaFiller, the known datasets of the five species for different parameter settings were used. The weights w_i for the component gap fillers and the threshold θ were searched for the parameter setting in order to obtain the highest F_2 score. As mentioned earlier, the parameters were searched in the range *(0,1)* with step-size of *0.01*.

We found that the optimal parameter setting was species dependent. So, we optimized the parameters for each species (dataset) separately as shown in Table 3.3. These were used as default settings for MeGaFiller on corresponding dataset. As can be seen, the *iIB711* dataset (the last one) gave different results from the other 4 datasets. So, we discuss results for the first 4 datasets. The optimal weights w_i for PFP-GF and B2G-GF were high while the optimal weights for EFICAz-GF were the lowest (except for the *iIB711* dataset). The optimal threshold θ ranged from 0.4 to 0.63. For each dataset, the optimal θ was quite close to the highest weight w_i. This suggests that the score of the highest weight component gap filler (usually PFP-GF) must be very close to 1 or it is made by at least two component gap fillers (e.g. Recall that EFICAz-GF scores are all 1.0, while PFG-GF scores range from 0.1 to 1). This result is consistent with our earlier stated objective of (a) leveraging the high recall of PFP-GF and (b) increasing the precision by having it "confirmed" by the other component gap filler.

Table 3.3 – Optimal parameters for MeGaFiller

Network dataset	w_1 (PFP-GF)	w_2 (B2G-GF)	w_3 (EFICAz-GF)	Θ
iIN800	0.60	0.10	0.30	0.63
iWV1314	0.43	0.47	0.10	0.40
iHD666	0.65	0.20	0.15	0.63
iMA871	0.40	0.50	0.10	0.40
iIB711	0.13	0.30	0.57	0.30

Note: values of parameters (w_i, θ) optimized by maximizing F_2 score on the known dataset N_K for each network. These weights and thresholds were used for gap filling on corresponding metabolic network.

3.3.4 Evaluation of MeGaFiller on known datasets

After parameter tuning, the optimized parameter settings were then used as default for MeGaFiller on the given dataset. The prediction results of MeGaFiller on known datasets for *iIN800* and *iWV1314* are shown in Figure 3.3. For both species, MeGaFiller produced more true positives than any one of its component gap fillers. In addition, the number of false positive has improved drastically as compared to PFP-GF.

For the yeast *iIN800* dataset, the number of true positives from MeGaFiller was 524, compared to 520 for PFP-GF, 445 for B2G-GF and 402 for EFICAz-GF. Additionally, the number of false positive was 154, drastically lower than 1,438 for PFP-GF, and comparable to those of EFICAz-GF (164) and B2G-GF (119). For the filamentous fungus *iWV1314* dataset, the number of true positives for MeGaFiller was 931, which was much better than 833 for PFP-GF, 844 for B2G-GF and 489 for EFICAz-GF. While the number of false positive went down to 979, compared with 2,137 for PFP-GF, 682 for EFICAz-GF and 119 for B2G-GF.

Figure 3.5 shows the F_2 score of MeGaFiller and the individual gap fillers, (PFP-GF, B2G-GF, and EFICAz-GF) on the known datasets of all five species. Clearly, MeGaFiller was consistently better than all of its component gap fillers over all the five datasets. In particular, we noted that the *iMA871* dataset, MeGaFiller had F_2 score of almost 60%, even when all the component gap filler had F_2 score below 50%. Summing over all the five species studied,

MeGaFiller achieved average F_2 score of 68%, with average recall of 73% and average precision of 55%.

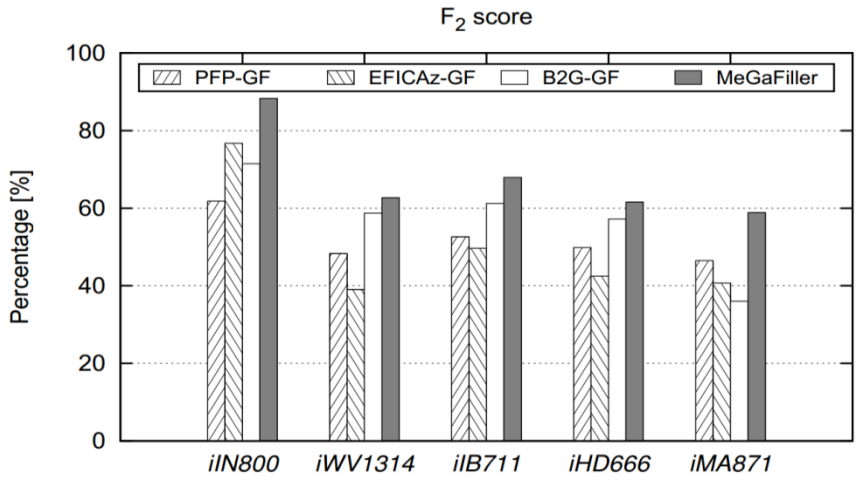

Figure 3.5 – Relative performance of different gap fillers and MeGaFiller

F_2 score is shown for each method on 5 network datasets. MeGaFiller achieved the highest value, which outperformed all components.

3.3.5 Comparing MeGaFiller with other variants of ensemble scheme

To further analyse the contribution of the weighted ensemble scheme used in MeGaFiller, we compared with two other ensemble variants. The first is a Non-Weighted ensemble in which the weights are equal (in this case, $w_i = 1/3$ for each of the 3 components). The second, called Common-2 ensemble, uses simple voting and keeps only predictions made by at least 2 component gap fillers (this version ignores all the weights and the scores in MeGaFiller).

We repeated the evaluation on these two ensemble variants and compared them with MeGaFiller. Table 3.4 shows the F2 score of the three ensemble variants. As expected, the three variants have very similar F2 scores. This table also shows that MeGaFiller (the weighted version) achieved the highest F2 score over all five datasets. The non-weighted ensemble variant performed only slightly better than the common-2 ensemble variant. We suggest that if no parameter tuning can be done, then the common-2 ensemble variant based on simple voting may also be a good strategy.

Table 3.4 – Performance of different variants of our ensemble gap fillers

Network dataset	MeGaFiller	Non-weighted	Common-2
iIN800	**88.27**	86.71	86.40
iWV1314	**62.67**	60.48	59.86
iHD666	**61.55**	58.77	58.39
iMA871	**58.86**	54.53	53.64
iIB711	**67.91**	67.14	66.91

Note: F2 score [%] of the weighted ensemble (MeGaFiller) is always better than that of non-weighted version and common-2 voted version. The non-weighted version was run the same as weighted version, except that the weights were fixed equally. The common voted version was run by taking only predictions that were made by at least 2 component gap fillers (ignoring both weights and scores). The largest value for each row is shown in bold. The non-weighted version slightly performed better than the common-2 voted version.

3.3.6 Comparing MeGaFiller with GFAOP and ADOMETA

MeGaFiller vs GFAOP

We compared MeGaFiller with an existing homology-based direct gap filling method, GFAOP. A direct comparison with GFAOP was not possible since the first step of GFAOP with identifying the protein family given the EC number requires expert domain knowledge which is not easy to automate in software. Instead, we compared the predictions of MeGaFiller against the set of metabolic gaps in *A. oryzae* that were previously filled by the GFAOP method. While this comparison is not ideal, it is a reasonably close approximation. GFAOP used older datasets than MeGaFiller, but GFAOP have domain expert input while MeGaFiller does not.

For this comparison, we first extracted the set of metabolic gaps from the *A. oryzae iWV1314* metabolic network that were previously filled by GFAOP, together with the set of gene-EC number pairs predicted by GFAOP. This set represents the "difficult-to-fill" metabolic gaps that had remained in the network before applying GFAOP, but were then successfully filled by GFAOP. We called this the recently-filled gaps dataset WV-RFG and it contains 162 gene-EC number pairs.

We used MeGaFiller to fill these metabolic gaps in WV-RFG. MeGaFiller managed to predict 169 gene-EC number pairs. These predictions were compared with WV-RFG (the

results obtained by GFAOP in [3]). MeGaFiller also predicted 102 (or 63%) of the 162 gene-EC number pairs filled by GFAOP. We note that this is a reasonably good performance by MeGaFiller since GFAOP uses domain expert input while MeGaFiller does not.

We then analysed the other 67 pairs predicted by MeGaFiller that were not in the WV-RFG. These predictions are either false predictions or additional gene-EC number pairs that were missed by GFAOP earlier. After our manual curation, we found that 38 (out of 67) pairs have strong supporting homology evidences over multiple annotation databases (e.g. CDD, Pfam, and UniProt databases). Thus, it is likely that these 38 pairs predicted by MeGaFiller are actually additional gene-EC number pairs for the *A. oryzae* metabolic network, but they were missed by GFAOP. In the following, we give two illustrative examples.

Example 1: Consider the endo-1,4-beta-xylanase reaction (EC:3.2.1.8) which is a metabolic gap in WV-RFG and is in the Polysaccharide metabolism. GFAOP predicted 6 gene-EC number pairs. For this reaction (EC:3.2.1.8), MeGaFiller gave a total of 8 gene-EC number pairs (including the 6 pairs predicted by GFAOP). The two additional pairs involve candidate genes with identifier AO090026000103 and AO090103000141 in *A. oryzae*. But, both these genes are currently un-annotated in the *A. oryzae* genome.

Through our manual curation, we found that both these candidates AO090026000103 and AO090103000141 matches (with e-value *2.3e-44* and *2.5e-44*, respectively) to Glycoside hydrolase (Glyco_hydro_11) domain in Pfam (PF00457) database. The family 11 of this domain comprises enzymes with only one known activity of xylanase (EC:3.2.1.8).

The previous method GFAOP missed both candidates. We found that there are two protein families (PF00457 and PF00331) that can perform this metabolic activity, and both are available in the Pfam previously. We conjecture that GFAOP probably took only one family (PF00331) as its input and hence missed these predictions.

Example 2: Consider the *A. oryzae* gene with identifier AO090009000675, which is currently annotated in DOGAN database as a putative sugar kinase, and assigned in the *A. oryzae iWV1314* network as a NADH kinase (EC:2.7.1.86, in NAD and NADP Conversion pathway). MeGaFiller predicted (by all component gap fillers) this gene for another metabolic

reaction NAD(+) kinase (EC:2.7.1.23). The Pfam homology confirms that it is a member of NAD(+) kinase family (PF01513) with an e-value (1.3e-47). This kinase family (PF01513) includes both EC:2.7.1.23 and EC:2.7.1.86 enzymatic functions. GFAOP found one of these, namely the EC:2.7.1.86 enzymatic function, while MeGaFiller found both of them. In this instance, MeGaFiller filled a gap and at the same time, found an additional metabolic reaction for an existing enzyme. This example shows that MeGaFiller can also supplement protein function predictions, and in this case, it improves the network for the NAD and NADP conversion co-factor pathway.

Explaining why GFAOP missed the remaining 61 metabolic gaps:

We analyzed why the GFAOP missed the 61 metabolic gaps in the *iWV1314* network. To fill a gap (given by EC number), GFAOP firstly needs to find specific protein family for the given EC number. Examining the 61 EC numbers of the gaps in the *iWV1314* dataset, we found that only 9 EC numbers (of the 61) have corresponding Pfam protein family translation. However, GFAOP could not find any protein encoding in the genome of *A. oryzae* that matches with these 9 protein families. The remaining 52 EC numbers cannot map into specific Pfam protein family (as already explained in Background section). Some of these are too general, some are mapped to many Pfam families, and some are grouped under complicated Pfam sub-domain structures. Thus, there is no single specific protein family that can be used by GFAOP for those gaps. In contrast, MeGaFiller was able to fill 12 of these 52 gaps in this category, as explained earlier.

MeGaFiller vs. ADOMETA

We also carried out a comparison of MeGaFiller with ADOMETA [90-92] which is a context-based method for gap filling. ADOMETA leverages gene association data [90-92] and can be used to predict new gaps as well as filling existing and predicting gaps. For comparison of MeGaFiller with ADOMETA, the published results of ADOMETA were used. The dataset used in ADOMETA was the metabolic network *iFF708* of yeast *S. cerevisiae* from year 2003. This dataset has 513 genes, 386 EC numbers, and 541 pairs. It was reported that during self-testing for ADOMETA with this *iFF708* dataset and combined with gene association data, achieved

60% recall based on their top-50 predicted candidates. It is noted that the precision of ADOMETA was not reported.

We ran MeGaFiller on the same *iFF708* dataset, and achieved a recall of 87% with a precision of 77%. These are significantly better results, in both recall and precision. Of course, this is not a completely fair comparison – part of the improvement could be due to the more up-to-date reference information used by the component protein function predictors. However, we believe that homology evidence (where they exist) is stronger than association evidence in predicting these candidate genes. By relying on homology evidence to make its prediction, we believe that the candidate genes predicted by MeGaFiller are more reliable.

This result also suggests that one reason MeGaFiller worked well for less-characterized genomes is that the homology reference for them, in other existing genomes, was richly available and these could help MeGaFiller and other homology based methods like GFAOP to find the correct candidate genes.

3.3.7 Effectiveness of MeGaFiller in filling metabolic gaps

We evaluated the effectiveness of MeGaFiller in filling the metabolic gaps in the metabolic gap dataset N_G for the five metabolic networks. For each metabolic gap from the N_G, MeGaFiller produced a list of candidate genes, if any, from the target genome that perform the function of the metabolic gap. We measured the number of gaps filled and the total number of candidate genes predicted for these gaps.

Table 3.5 shows the results obtained by MeGaFiller. For each network, it shows the number of metabolic gaps, the number of gaps filled (putatively with at least one candidate gene), the total number of candidate genes predicted for these gaps, and the percentage of gaps putatively filled by MeGaFiller. For the *iIN800* network, MeGaFiller putatively filled 15 out of 52 gaps (29%), and for *iWV1314* network, it putatively filled 12 out of 61 gaps (20%). It obtained even better results for two of the less well-studied species which were 37% for *iIB711* network and 61% for *iHD666* network. On average, MeGaFiller putatively filled 35% of the

metabolic gaps in the five networks. In following description, we highlight some results from MeGaFiller.

Table 3.5 – Number of putatively filled metabolic gaps for the five metabolic networks

Network dataset	iIN800	iWV1314	iHD666	iMA871	iIB711
Number of metabolic gaps	52	61	28	89	68
Number of metabolic gaps putatively filled	***15***	***12***	***17***	***23***	***25***
Number of putative candidates	***25***	***15***	***68***	***61***	***64***
Percentage of putative metabolic gap filled	29%	20%	61%	26%	37%

Note: For *A. oryzae* metabolic network (*iWV1314*), MeGaFiller predicted one or more candidate genes for 12 (out of 61 (20%)) metabolic gaps (and a total of 15 candidates).

Example 3*:* Consider the fumarylacetoacetate hydrolase reaction (EC:3.7.1.2) in Phenylalanine, tyrosine, and tryptophan biosynthesis pathway in the *S. cerevisiae iIN800* network. This is currently a metabolic gap in the *S. cerevisiae iIN800* network. MeGaFiller predicted the candidate gene identifier YNL168C in *S. cerevisiae* for this reaction (EC:3.7.1.2). Through our manual curation, we found that YNL168C matches to the Pfam FAA_hydrolase family (PF01557) with an e-value of 1.1e-49, and significantly matches with InterPro entry IPR002529 (fumarylacetoacetase, EC:3.7.1.2). Hence, there is direct evidence to support this candidate gene, even though it is currently still unknown function in the SGD database.

Example 4: In *A. nidulans iHD666* network, the reaction EC:3.1.3.3 (phosphoserine phosphatase) is a metabolic gap. MeGaFiller predicted AN10593 is a candidate gene for this reaction. Currently, in AspGD database, this gene is annotated as uncharacterized function. In fact, this candidate gene hits to HAD family in Pfam database with significant e-value of 2.6e-16. This family involves phosphoserine phosphatase activity. Besides, sequence similarity searching against Swiss-Prot database also gives supporting evidence for this candidate gene.

Example 5: In *A. niger iMA871* network, the reaction EC:4.2.3.5 (chorismate synthase) is a metabolic gap. MeGaFiller found the gene ID 54235 as a candidate in *A. niger* genome. This candidate matches well with the chorismate synthase domain with e-value of 6.5e-130 in Pfam database.

Example 6: In *S. coelicolor iIB711* network, the reaction EC:4.1.1.36 (phosphopantothenoylcysteine decarboxylase) is a metabolic gap. MeGaFiller predicted SCO1477 is a candidate gene. This candidate is annotated as putative flavoprotein homologue in UniProt database. However, it matches well with DFP (with e-value of 1.6e-69) and Flavoprotein (with e-value of 8.6e-34) domains in Pfam database. Furthermore, KEGG database also confirms EC:4.1.1.36 activity for this candidate gene.

In addition, we further analysed which component gap fillers predicted the most gaps. As expected, within MeGaFiller, PFP-GF always produces the most number of candidate genes predictions. For examples, for the *S. coelicolor iIB711* dataset, PFP-GF predicted 63 out of the 64 candidate genes from MeGaFiller, while B2G-GF predicted 60 candidate genes and EFICAz-GF predicted 9 candidate genes. Overall, within MeGaFiller, the general gap fillers (PFP-GF and B2G-GF) always contribute more predictions than the enzyme-specific gap filler (EFICAz-GF).

3.3.8 Filling critical gaps in metabolic network of *A. oryzae*

A more detailed analysis of the metabolic network *iWV1314* for *A. oryzae* shows that there are 61 metabolic gaps (EC numbers) involved in 65 reactions that are spread over 37 metabolic pathways. To judge whether a gap is critical, we manually examined its reference pathway map given by KEGG database. A reaction in a pathway is called critical if it is the only reaction that consumes/produces a metabolite that is specific to the pathway. In other words, without that transformation, this pathway will not be connected, and hence the reaction is critical.

There are 28 critical gaps in the *A. oryzae iWV1314* network, and 33 non-critical gaps. Significantly, MeGaFiller predicted 12 candidate genes for 10 of these critical gaps. This means that MeGaFiller filled the gaps that are most likely to improve the connectivity of metabolic networks.

Example 7: Consider the pantetheine-phosphate adenylyltransferase (PPAT) reaction (EC:2.7.7.3) which is currently a metabolic gap in the *A. oryzae iWV1314* network. This reaction is a critical gap in the Coenzyme A and pantothenate biosynthesis pathway (see Figure

3.6) as it is the only transformation that produces Dephospho-CoA, which is the substrate to produce Coenzyme A. Without this reaction, the pathway is not functional and the Coenzyme A cannot be synthesized by this pathway.

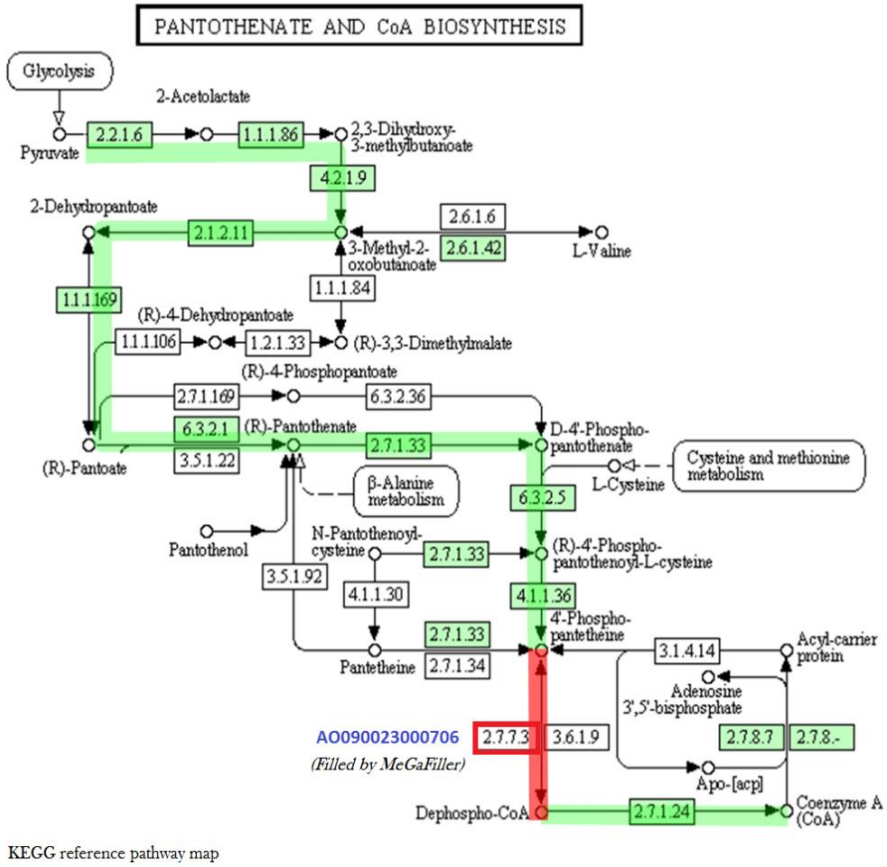

Figure 3.6 – Filling gap in Pantothenate and CoA biosynthesis pathway for A. oryzae

The picture was modified from KEGG Pantothenate and CoA biosynthesis pathway (aor00770). Green-filled boxes are reactions with already identified genes in A. oryzae. White boxes are reactions without genes identified in A. oryzae. The EC:2.7.7.3 reaction (thick red-border box) is the "bottle-neck" for producing Dephospho-CoA, the substrate metabolite for CoA synthesis. MeGaFiller predicted AO090023000706 is the protein that catalyses for this reaction in A. oryzae.

This PPAT gap reaction by EC:2.7.7.3 was predicted by MeGaFiller (both PFP-GF and B2G-GF) to be catalysed by the candidate gene with identifier AO090023000706 in the *A. oryzae* genome. But, currently this gene shows un-annotated function in the DOGAN database.

However, we found strong supporting evidence for this prediction. Firstly, the protein matches with PPAT_CoAS (phosphopantetheine adenylyltransferase domain) in CDD database with an e-value of *1.48e-36*. It also matches to Pfam's cytidylyltransferase domain (which is

more general than PPAT) with an e-value of *6.1e-05*. Another matching CDD entry is PRK01170, which is provisionally annotated as phosphopantetheine adenylyltransferase/ unknown domain fusion protein. In addition, the corresponding ortholog in yeast (assigned by Ortho-MCL database) is the gene YGR277C, which is annotated as a PPAT by SGD database. With these strongly supporting evidences, we believed that AO090023000706 is the missing gene for the reaction with EC:2.7.7.3. With the assignment of the gene AO090023000706 to this reaction, the pathway becomes complete function.

3.3.9 Putative enhancement

While MeGaFiller was designed primarily to fill metabolic gaps, we can also use it as a method to make putative enhancement to current metabolic networks. This enhancement is in the form of (a) putative candidate genes for existing reactions in the network, and (b) novel putative reactions for the current metabolic network. Here, we give some results.

Novel Candidate Genes: To do this for any target species, we ran MeGaFiller on the genome of the target species using the list of all EC numbers from the metabolic network of the species. We then filtered out the predictions that were already found in the networks. The remaining predictions contain novel candidate genes for existing reactions in the network.

Table 3.6 – Novel candidate genes predicted for the five metabolic networks

Network dataset	*iIN800*	*iWV1314*	*iHD666*	*iMA871*	*iIB711*
Number of genes in current network	707	1346	674	831	711
Number of novel candidate genes	***231***	***587***	***384***	***289***	***280***

Note: The number of novel candidate genes predicted by MeGaFiller for the five metabolic networks. These candidates need to be further curated, but they represent big potential enhancement in the gene coverage of these metabolic networks.

We ran this for all the five networks and the results are shown in Table 3.6. For the *A. oryzae iWV1314* network, MeGaFiller predicted 587 novel candidate genes (for 215 EC numbers). The numbers of novel candidate genes for *S. cerevisiae iIN800*, *A. nidulans iHD666*, *A. niger iMA871* and *S. coelicolor iIB711* networks were 231, 384, 289 and 280, respectively.

These predicted candidate genes need to be further curated and validated, but they give a valuable supplement of candidate genes for enhancement of these metabolic networks.

Novel Putative Metabolic Reactions: To use MeGaFiller to predict novel metabolic reaction for current networks, we first retrieved all the EC numbers in the reference metabolic pathway (with identifier ec01100) from KEGG database. In all, there were 1,464 EC numbers relevant to metabolism. We filtered all EC numbers that were already found in the *A. oryzae* network (*iWV1314*). The remaining 753 EC numbers were used as input for MeGaFiller. Of these, MeGaFiller predicted 369 candidate genes for 119 new EC numbers with corresponding novel putative metabolic reactions for *A. oryzae*. These novel putative reactions also need to be further curated and validated. Potentially they can further enhance the metabolic network.

3.4 Discussion and conclusions

Metabolic gaps that exist after network reconstruction are usually “difficult-to-fill”, since earlier gap filling methods have already failed to find them. While previous homology methods for gap filling that are based on protein family profile have been successfully used to enhance the reconstructed networks [3, 88], they may fail if the protein family is poorly defined. Our approach based on retrofitting protein function prediction has indeed overcome the issue, since it does not require the concrete protein family, any individual protein in reference databases could help. We demonstrated in this work that retrofitting state-of-the-art protein function predictors can help to find candidates for “difficult-to-fill” local metabolic gaps that were missed by previous direct gap filling methods. We implemented and tested an ensemble MeGaFiller method, which rationally combined three retrofitted gap fillers. We also performed gap filling and manual curation for *A. oryzae* network, and putative enhancement for the other four reconstructed metabolic networks. The fact that MeGaFiller was able to fill 12 gaps (missed by previous method GFAOP for *A. oryzae* network), which have no specific protein family interpretation, has confirmed our idea. Furthermore, MeGaFiller is able to predict many additional candidate genes for existing reactions, as well as novel putative metabolic reactions.

Re-validation on filled gaps in *A. oryzae* network and manual inspection showed that our method was able to reliably propose more candidate genes that were missed by previous methods. There are strong supporting evidences found for these candidate genes in *A. oryzae* metabolic network, which suggests that our methodology is reliable. Thus, our method can serve as an effective bioinformatics tool for filling metabolic gaps and enhancing reconstructed metabolic networks.

Chapter 4

EnzDP: improving enzyme annotation by domain composition profiles

4.1 Introduction

An essential step for genome-scale metabolic reconstruction [13, 87] of a cell is to completely determine its enzymatic activities. The quality of this step strongly affects the quality of the reconstructed metabolic networks, especially by improving enzyme coverage and reducing missing genes/reactions [11, 12]. Essentially, the first step of network reconstruction is enzyme annotation, whereby enzymes are assigned 4-digit EC numbers to describe their metabolic functions, up to substrate-binding level. After that, reconstruction continues with network assembly, gap filling and network enhancement [17]. However, gaps (or holes) [90] that appear in later steps, become difficult to fill without revising the enzyme annotation procedure. It was shown in [104] that many difficult-to-fill gaps in *A. oryzae* network [3] can be filled by retrofitting function annotators with a better gap-finding strategy. Thus, gap filling and network enhancement can be done by improving enzyme annotation.

In fact, reconstructed metabolic networks still contain gaps, as well as have low enzyme coverage. For example, in the reconstructed networks of 5 species under a study in 2014, 6-19% of unique reactions are still gapped [104]. The presence of such a significant number of gaps implies that enzyme annotation in the first step was far from complete. Therefore, there is a need for new and robust idea to improve enzyme function classification quality.

The problem of ***enzyme function prediction*** is well established [45, 105] , and 3 main approaches have been attempted: network-based, structure-based and homology-based. Chapter 2 gives a detail review on these approaches.

Network-based and structure-based methods have low coverage of predictable enzyme families. Homology-based approaches may get high coverage, but its accuracy can be further improved. A bottom-up homology-based approach can improve coverage by re-building HMM profiles for all known enzymes. Briefly, training enzymes are clustered into subgroup families, and then each subgroup is represented by a sequence profile.

A common issue with bottom-up methods is that, the clustering procedure relies firmly on the BLAST similarity score, ignoring information about protein functional domains and domain patterns. These ultimate features are conserved regions of proteins, which well characterize enzyme functions. Cai et al. [106] effectively used domain architectures for predicting first digit of EC numbers. This gives a hint for improving enzyme annotation with the use of domain architecture.

In this research, a stringent protocol for enzyme annotation is presented. Our protocol integrates a novel idea of weighted protein domain architectures into bottom-up enzyme profile building. Specifically, we used domain architecture to score the association between domain families and enzyme families (this score is called DEAS). Then this score is used in the clustering step, instead of using BLAST similarity score as traditional methods do. We further improved the enzyme annotation protocol with a stringent classification procedure, including optimal threshold setting, overlap checking and active site checking. Our EnzDP protocol gives better enzyme coverage with higher prediction accuracy, compared to other alternatives.

4.1.1 Pros and cons of current approaches

Network-based methods work without using genome sequence data. They are thus able to predict the functions of novel genes (i.e. functions that have no known homologs). However, the significant limit is that this approach cannot infer functions that are more specific than functions of genes in their network neighborhood. Furthermore, since these methods use gene

association data, which is "weaker" than homology data, both the coverage and accuracy are worse than direct homology-based methods.

Structure-based approaches make use of 3D structures, which are more specific and accurate indicators for determining biological functions. However, the current number of known 3D templates is limited. For example, the PDB, as of June-2014, contains domain structures that cover about 2900 enzyme families, which is only about 60% of known enzyme families annotated in Swiss-Prot. Therefore, structure-based approaches have low coverage.

Conserved protein/family domains are reliable features for function identification. Despite that, methods that directly map domain identifications to enzyme function annotations have low prediction coverage. For example, the number of predictable EC numbers using PROSITE is less than 31% of all known EC numbers. The PF2GO2EC mapping also has very low coverage (13%) and may lead to misleading or incorrect assignment of enzymatic functions. The reason of these incorrect assignments is that, one enzyme family can carry multiple different domains, and at the same time, one domain is carried by different families, (see Figure 4.1 for example).

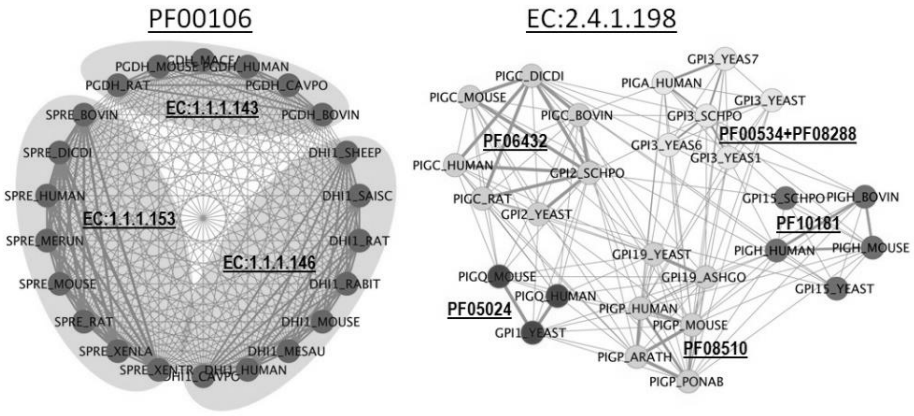

Figure 4.1 – Example of enzymes and domains

Nodes represent enzyme proteins. Edge thickness represents sequence similarity (Blast e-value) between nodes. *Left*: Domain PF00106 (*short_adh*) is carried by different enzyme families (only 3/90 families is shown). *Right*: Enzymes in EC:2.4.1.198 carry 6 different domains..

Figure 4.1 shows examples of difficult cases. The domain *short_adh* (PF00106) (on the left) is carried by 700 sequences which belong to 90 different enzyme families (i.e, 90 enzyme numbers). Thus, sequence/pattern similarity may match a protein carrying this domain to any of those families, which is ambiguous. This can be overcome by using different thresholds for different families/sub-families individually. On the other hand, the EC:2.4.1.298 enzyme family contains 5 sub-families, each carries different set of domain families. Thus, to describe this family, multiple profiles should be used.

To infer enzymatic functions from protein domain identifications, the association score between domain families and enzyme functions can be used. This association score (we called DEAS – see Methods) can be directly computed from Swiss-Prot database. Analysis shows that, DEAS can improve either coverage or accuracy, but not both.

Bottom-up methods can improve both coverage and accuracy over other methods. However, in current annotation protocol, the clustering step is performed using Blast sequence similarity score, which can be improved by the above novel idea of domain enzyme association score. Furthermore, the threshold setting in the common protocol is too simple, which can be replaced by a more stringent strategy for each sub-family individually.

4.1.2 EnzDP protocol

In this work, we improve the common enzyme annotation protocol with (1) a stringent strategy for subgroup clustering, (2) a calibrated optimal threshold setting, and (3) an efficient classifying procedure.

For the *first improvement*, our clustering strategy explicitly utilizes the weighted domain architectures of enzymes (DEAS) to cluster them into proper subgroups. Previous methods use whole-sequence similarity for clustering, while we focus on combination of conserved regions. This use of domain architectures is helpful for the cases of proteins with multiple domains, which are not rare in Swiss-Prot database. It was previously shown that protein domain architecture can be effectively used for clustering highly similar orthologous subgroups and improves BLAST-based ortholog assignment [107, 108]. Cai et al. [106] reported an overall

85.5% success rate in classifying first EC number digit for sequences of low identity, using domain architecture. In our work, we use weighted domain architecture for classifying enzymes up to all 4 digits.

Due to fact that mis-annotations are common in public databases [109], training sequences should be carefully filtered. Furnham et al. [109] showed that, compared to other public databases, Swiss-Prot/Uniprot has the least number of mis-annotations. In addition, Uniprot has a large amount of constantly updated genomics annotation data. Our method retrieves high-quality training sequences from Swiss-Prot, since it is a large manually reviewed portion of Uniprot.

In our approach, two training enzyme sequences are clustered into the same enzyme subgroup if: *(a) they have the same EC number annotation*, and *(b), they carry two sets of functional domains that are highly similar*, where similarity is weighted by the domain enzyme association score DEAS (see Methods). After that, a Hidden Markov Models (HMM) profile is built, with an optimized threshold, for each enzyme subgroup. The probabilistic HMM profiles [110] may be considered a better alternative to traditional PSSM profiles (used by PRIAM), especially for long domains/regions.

For the *second improvement*, each profile is calibrated for an optimal cut-off threshold, based on maximizing F1 score in a strict 5-fold cross-validation procedure. Thus, profiles for the subgroups of each EC number may have different thresholds, which can discriminate members of different enzyme families. This is different from the PRIAM method, as in 2003 [36], PRIAM uses a fixed optimized threshold (of 1e-30) for all profiles.

The *third improvement* of our method is attributed to an efficient classification procedure. Firstly, the method classifies multi-enzymes (enzymes catalyzing multiple reactions) first, and then mono-enzymes (enzymes catalyzing for single enzymatic reaction) separately. Secondly, the classification procedure focuses on accurately rejecting false hits by evaluating the overlap of matching regions. Lastly, we also check for the set of known residues (active/binding sites) specific to each enzyme subgroup. The final prediction is assigned a relative precision score as the likelihood/reliability of the prediction.

Our analysis showed that, with this stringent protocol, our novel method achieved a high enzyme coverage, up to 90% of known annotated enzymes families in Swiss-Prot. EnzDP got a high accuracy (94.5%) in a *solid* 5-fold cross validation, and outperformed state-of-the-art methods in many experimental tests, including leave-one-out cross validation on several representative genomes (a variation of jackknife test) and validations on independent datasets of recently annotated enzymes. It is also a fast automatic enzyme annotator, which could annotate enzyme content for a metagenome of human oral microbes (500MB data) in few hours. Thus, it can be used for timely interpretation of whole-genome and metagenomics data.

4.2 Methods

4.2.1 Computing DEAS

The weighted mappings between Pfam domain identifier and EC numbers was derived based on the frequency of their co-occurrences in manually curated annotation of Swis-Prot database. We called it domain-enzyme association score, DEAS.

If a protein carries a domain f_y and is annotated by an EC number e_x, we say e_x and f_y are co-occurrence. The DEAS is calculated using Sørensen–Dice index by following formula:

$$DEAS(f_y, e_x) = \frac{2 * n(f_y, e_x)}{n(f_y) + n(e_x)} \qquad \textbf{(1)}$$

where: $n(f_y)$ is the number of proteins that carry the domain f_y ; $n(e_x)$ is the number of proteins that annotated by the EC number e_x, and $n(f_y,e_x)$ is the number of proteins that carry domain f_y and are annotated by the EC number e_x.

Thus, each pair of the mappings is assigned a DEAS score. A higher score means a stronger association. Thus, a pair with score of 1.00 has 100% co-occurrence. This weighted mapping can be used with user predefined cut-off.

4.2.2 EnzDP Protocol

The training protocol in EnzDP contains 4 main steps (Figure 4.4), as follow.

Training

In the first training step, protein sequences and annotations were retrieved from Swiss-Prot database. Enzymes and non-enzymes were separated into different sets. Known active site information, namely the active site amino acid and its position, was also retrieved.

In second step, enzyme clustering was performed. Enzyme domain architecture was used for clustering purpose. Domain architecture of an enzyme is the set of functional domains that it carries. Two enzyme sequences are clustered into the same subgroup if the two conditions below are satisfied at the same time:

(a) They have the same EC number annotation, and
(b) Their domain architectures are highly similar, where the similarity is measured by a weighted *Sørensen–Dice index*, as in formula (2).

Denote the domain architecture of a protein P by $P = (p_i)$, where each p_i is a functional domain carried by P. The similarity of domain architectures of two proteins P and Q, for a given EC number e, leverages on their common domains, is calculated by the following formula (which can be seen as a weighted version of Sørensen–Dice index):

$$sim(P, Q|e) = \frac{2 * \sum_{f_i \in P \cap Q} DEAS(f_i, e)}{\sum_{p_i \in P} DEAS(p_i, e) + \sum_{q_i \in Q} DEAS(q_i, e)} \qquad \textbf{(2)}$$

where DEAS was calculated by formula (1).

For the second condition being satisfied, the similarity must be higher than a specified threshold, i.e., $sim(P,Q|e) \geq \theta$. In our setting, θ was chosen as 2/3.

Example

Consider 3 proteins P, Q, and R of the same EC number e, with domain content as $P = (d,b)$, $Q = (d,c)$, $R = (b,c)$. For non-weighted version (all *DEAS* component in the above formula equals to 1), any pair of these proteins will have similarity of 1/2. However, suppose the domain d is associated with the enzyme e much more stronger than other domain b and c, with following scores: $DEAS(d,e) = 0.5$, and $DEAS(b,e) = DEAS(c,e) = 0.1$. Then, these similarities become: $sim(P,Q|e) = 5/6$; $sim(P,R|e) = sim(Q,R|e) = 1/4$. In this case, the pair (P, Q) has much higher

similarity, since both the proteins carry the *strong* domain *d*. Note that, if two proteins have the same domain contents, then they will have 100% similarity. Thus, if this enzyme family has another protein *S* with the domain content *S = (b,c)*, the similarity between *(R,S)* will be 1.

Furthermore in this step, if an enzyme sequence cannot be grouped with any other enzymes, it forms a singleton subgroup. As of Jan-2014, there are 1.22% (3083/ 253797) of known enzymes are in singleton subgroups. Thus, ignoring these enzymes has negligible effect. However, there are 461 (11.2%) of 4126 EC numbers that have only 1 enzyme sequence annotated. Thus, these 461 singleton subgroups are representative. We treated all representative subgroups for these 461 single-sized enzymes, and any other subgroup of size 2 or more as primary. While, remaining 2667 singleton subgroups were treated as optional.

In the third step, a hidden Markov Model (HMM) profile was built for each enzyme subgroup. First, the training sequences were aligned using multiple sequence alignment (MSA) software Clustal-Omega [111]. Then, from the MSA, a HMM profile was built using hmmbuild program in HMMER3.0 package [110]. At the end of this process, raw HMM profiles were obtained.

Profile threshold calibration

In the last step, profile threshold calibration was performed. We used the following rule to set the optimal threshold, in which the F1-score (harmonic mean of recall and precision) is optimized. First, the profile is used to search against Swiss-Prot database. True hits (*hits that are true family members*), false hits (*hits that are not family members*) and their corresponding bit-scores were recorded. After that, for each possible cut-off threshold among those recorded scores, the number of true positives, false positives, and corresponding F1-score were calculated. Next, the smallest cut-off score S that maximizes the F1-measure was calculated:

$$S = argmax\ F1(s)$$

Then, the optimal threshold S^* was set by: $S^* = (S+S')/2$, where S' is the next score after S in sorted decreasing order of recorded scores.

Significantly, the threshold calculation for each subgroup of each EC number was averaged after a *k-fold* cross validation procedure, as following. For each EC number subgroup, the training sequences were equally randomly split into *k* parts. Each round, *(k-1)* parts of enzyme sequences were used for training, while the remaining part was set hidden from training and used in testing. The optimal *S** is averaged over a total *10*k* rounds. Only EC numbers with at least *3* sequences were run through cross validation. We set *k = 5* for EC numbers with 5 or more sequences, and *k =3* (or *4*), for EC numbers with *3* (or *4*) sequences.

Finally, the calculated scores s and their corresponding precisions pre(s) were stored together with the profiles as a library of enzyme profiles.

Classification

Enzyme classification procedure in EnzDP protocol (Figure 4.4, right) is following. Firstly, the profile library was scanned against the input sequences, using hmmsearch program in the HMMER3.0 package. The raw hits with corresponding bit-scores of at least 0.05 were recorded. After that, each hit was analyzed and a score r was assigned to it by following description. First, a precision measure pre(s) was calculated for s, based on the recorded scores of true and false hits that were stored together with the profiles. Then, the relative precision r for a prediction with score s is calculated as follows:

$$r_s = min\left\{1, \frac{pre(s)}{pre(S^*)} * \frac{s}{S^*}\right\}$$

where, *S** is the optimal threshold of the profile. Note that, if the hit score is greater than the optimal score, and the optimal score has 100% precision, then the relative precision of the hit is set as one *(rs = 1).* Otherwise, it is scaled down relatively, comparing to the optimal threshold.

Secondly, the list of candidates was filtered, as follows. All hits with scores lower than a user specified threshold were discarded. After that, if the hit enzyme subgroup belongs to a multi-enzyme family, EnzDP accepts it immediately as a candidate. If two different candidate subgroups do not belong to a multi-enzyme family, then EnzDP looks at the two regions on the

input sequence that hit by the two profiles. If these two regions overlap by more than 20% of their combined length, the hit with smaller score is rejected. At the end, candidates that remain after filtering are collected as final predictions.

On the other hand, hits from the first step above will also be analyzed for match active sites, if active site information for the profile is available. This is done by analyzing the alignment between input sequence and hit profiles, and looking for match residues and match positions. The match active site information was stored separately for manual inspection.

4.2.3 Evaluation Criteria

For each test, the number of true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) was counted. Following measurements were calculated to evaluate the prediction performance.

$$\text{Recall} = TP / (TP + FN)$$

$$\text{Precision} = TP / (TP + FP)$$

$$F1 = 2 * \text{Recall} * \text{Precision} / (\text{Recall} + \text{Precision})$$

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

For cross validation over N enzyme families, micro and macro performances were calculated as following.

$$Micro\_Recall = \frac{\sum_{i=1}^{N} Rec_i}{N}$$

$$Micro\_Precision = \frac{\sum_{i=1}^{N} Pre_i}{N}$$

$$Micro\_Accuracy = \frac{\sum_{i=1}^{N} Acc_i}{N}$$

$$Macro\_Recall = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FN_i}$$

$$Macro\_Precision = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FP_i}$$

$$Macro\_Accuracy = \frac{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} TN_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} TN_i + \sum_{i=1}^{N} FP_i + \sum_{i=1}^{N} FN_i}$$

where *TPi*, *FPi*, *TNi*, *FNi*, Rec_i, Pre_i and Acc_i are *TP*, *FP*, *TN*, *FN*, *Recall*, *Precision*, and *Accuracy* for the test on the family i^{th}, respectively.

Leave one-out cross validation

This validation was done by excluding one whole genome at a time from training data and using it for testing. We performed leave one-out cross validation (LOOCV) for several genomes, namely *E. coli*, *B. aphidicola*, *P. falciparum*, *H. influenzae*, and *M. genitalium*. These species was chosen for testing since their genomes are completely annotated in the benchmark Swiss-Prot.

Five-fold cross validation settings

To evaluate EnzDP, we performed a solid 5-fold cross validation for each EC number that has at least 3 sequences annotated. Overall, the number of EC classes have been validated is 2619 (from Swiss-Prot release Sep-2013, excluding EC numbers of less than 3 EC digits). For EC number of 3 (or 4) sequences, we performed 3-fold (or 4-fold) validation only. For each EC number, we randomly split the enzyme sequences in 5 parts. Each round, we used 4 parts for training and the remaining part for testing. This imitates the situation in which only 80% of the enzymes is known, and used to predict the unseen 20%. More rigorously, each testing set also contains the false sub-family members that best hit to the profiles. These sub-family best hits were not used in the threshold calculation during training. The threshold calculation process did not make use of the information whether a hit is true sub-family member or not, it only classified between true family member and false family member. Furthermore, the number of false-members was selected equal to the number of correct members to avoid bias towards true negatives. Finally, the average performance was calculated over all rounds of cross validation.

4.2.4 Data preparation

Training data was retrieved from manually curated Swiss-Prot database. Protein sequence, enzyme annotation, Pfam domain identifier, and active site information were collected. For the release of Swiss-Prot on January-2014, there are 542258 proteins, in which 253797 are enzymes

(46.8%). Among these enzymes, 242081 (95.4%) are mono-enzyme (enzyme annotated with 1 EC number), and 11716 (4.6%) multi-enzymes. All these 253797 enzymes were annotated to 4216 EC numbers, to form 268282 EC number – protein identifier pairs. Similar statistics for other releases are shown in Table 4.1.

On the other hand, there is a total of 89311 proteins have known active site. Among those, 3008 proteins are non-enzymes, 80672 mono-enzymes and 3631 multi-enzymes. Enzymes with active sites belong to 1870 different EC numbers. This active site information (namely, the amino acid of the site and its position in the enzyme) was used to map into the corresponding profile. The sites and theirs mapped position were stored together with the profile.

Table 4.1 – Statistics of enzyme datasets from Swiss-Prot

	#EC numbers	*# enzymes*	*# pairs*
Oct 2008	2610	169192	176823
Jan 2011	3217	244680	256929
Jan 2014	4126	253797	268282
Jan14-Oct08	1252	60024	62500
Jan14-Jan11	586	2463	2591

Note: Statistics of Swiss-Prot database is showed at different release timestamps: Jan14-Oct08 (Jan14-Jan11) denotes data newly added from Oct-2008 (Jan-2011) to Jan-2014.

To evaluate methods, we used newly added sequences between different releases of Swiss-Prot. Since the training data of EFICAz was Jan-2011 and of CatFam was Oct-2008, we retrieved data at these timestamps to train our method. After that, we selected sequences that exclusively belong to Jan-2014 release for testing. Furthermore, EC numbers that were un-predictable from training dataset were excluded. For example, from Jan-2011 to Jan-2014, there are 909 = (4126-3217) EC numbers newly added. These new EC numbers were newly found after Jan-2011, and thus were excluded since all methods cannot predict them. The testing enzyme datasets, Jan14-Oct08 and Jan14-Jan11, were obtained as shown in Table 4.1. Note

that, we excluded enzymes with incomplete EC number annotation, which are those of less than 3 known digits.

4.3 Results

4.3.1 Domain-Enzyme Association Scoring (DEAS)

To derive a direct mapping between domain identifiers and EC numbers, the domain enzyme association score, DEAS, was computed (see Methods). This association score leverages on the frequency of the co-occurrence between domain identifiers and EC numbers in all manually curated Swiss-Prot database entries. The DEAS score is directly proportional to the co-occurrence frequency, and inversely proportional to the sum of individual occurrences. A higher DEAS value implies a stronger association. If an EC number and a domain always appear together, its DEAS is 100%. The DEAS not only provides mappings between EC numbers and domain identifications, it also gives a weight for each pairing. Therefore, the mapping can be used with different user-predefined cut-off thresholds.

A 5-fold cross validation procedure was performed to evaluate the accuracy of this mapping. For each of 5 rounds, 1 out of 5 equal random parts of Swiss-Prot proteins was set hidden from training and used for testing. The statistics were averaged for each EC number first, then for all predictable EC numbers. Figure 4.2 shows its performance against different cut-off thresholds.

As can be seen, there is a trade-off between coverage and performance of DEAS. While the precision gradually increased with the increasing of cut-off, the coverage decreased sharply. DEAS precision increased from 50% to 100%, but its coverage decreased from 2526 predictable EC numbers down to 220 (more than 11 times lower). The 220 EC numbers with DEAS score of 1 is only 5.2% of all predictable EC numbers. This means that there are very few enzyme families that have *unique* functional domains associated.

Figure 4.3 shows the intersection between PF2GO2EC (a combination of EC2GO and Pfam2GO mappings) and DEAS-50 (i.e. DEAS with cut-off 0.5). They share 406 mappings.

Those common mappings have higher recall of 88%, precision of 92%, over 347 predictable EC numbers.

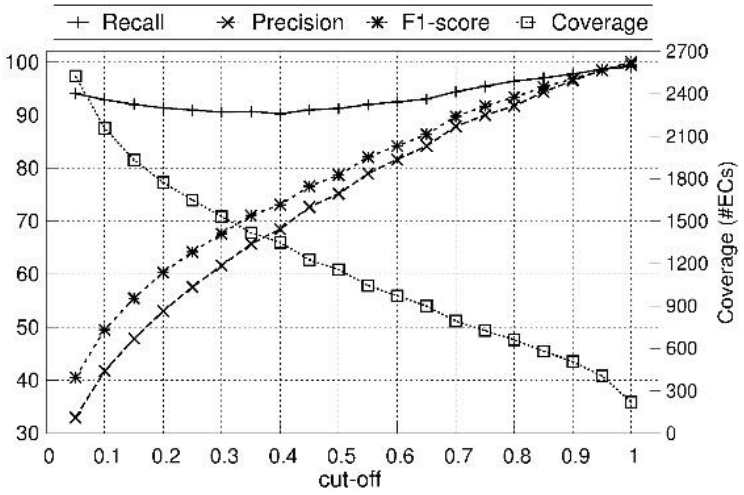

Figure 4.2 – Coverage and performance of DEAS with varying cut-off thresholds

Coverage is the number of predictable EC numbers. Performance is shown in percentage

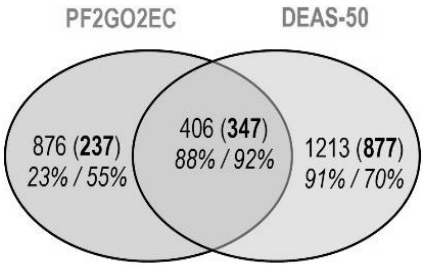

Figure 4.3 – Intersection of DEAS and PF2GO2EC mappings

They have 406 mappings in common, which covers 347 EC numbers. These common mappings achieved 88% recall and 92% precision, over those 347 predictable EC numbers.

The agreement between the two mappings is low. The intersection covers only 16% of the union. These common mappings are the most accurate, compared to the non-common mappings. Significantly, the DEAS exclusive mappings had much higher average recall (91%) and precision (70%) than PF2GO2EC exclusive mappings. This indicates that our mapping is more accurate than PF2GO2EC. However, both mappings can still be improved.

4.3.2 EnzDP protocol

EnzDP protocol leverages on clustering enzyme subgroups by weighted domain architectures, scored by DEAS. Training protocol and classification protocol are shown in Figure 4.4 (see Methods section for more details.)

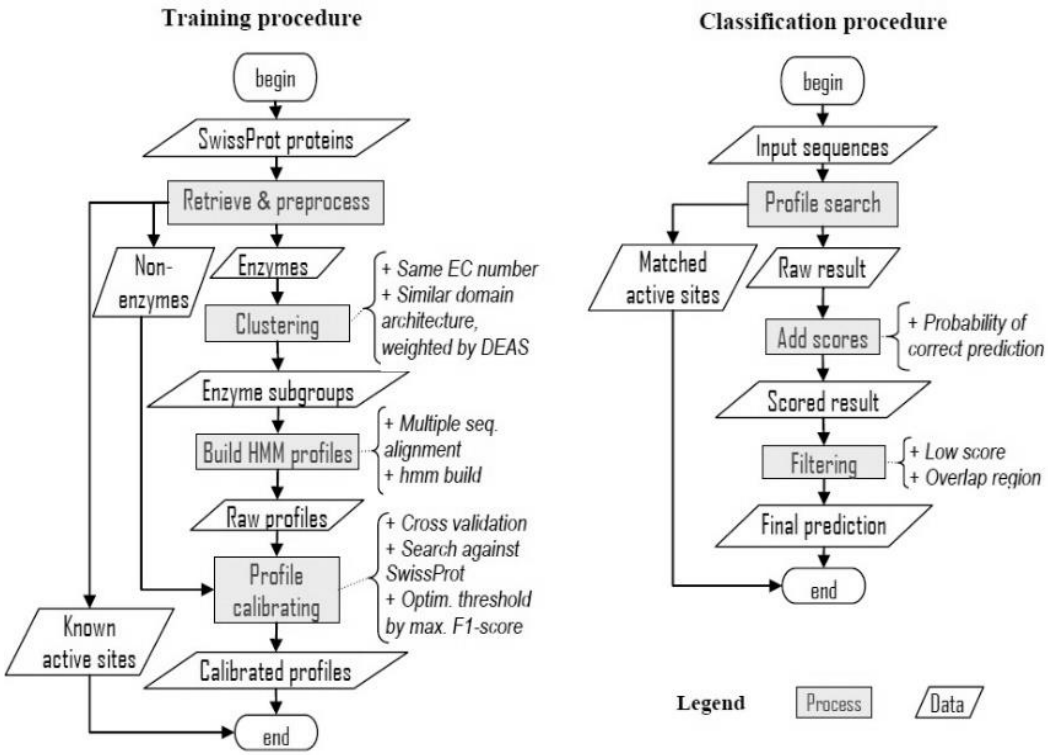

Figure 4.4 – EnzDP protocol: (left) profile training and (right) enzyme classifying

For the Oct-2014 training data, EnzDP built a library of 9501 profiles. Amongst those, 6834 profiles are primary and 2667 are optional (built from 1 enzyme sequence.) Among primary profiles, 5181 (76%) were built from at least 3 sequences, which map to 2619 EC numbers. The number of predictable EC numbers is 3152 (with 6373 profiles) when considering profiles trained from at least 2 sequences.

Table 4.2 shows a performance and coverage comparison between EnzDP with different top-down methods. As can be seen, EnzDP achieved the highest F1 score (more than 95%). This significantly outperformed top-down methods. Furthermore, our method had much higher coverage. For example, EnzDP can predict for 3152 EC numbers, which is 2.4-fold higher than

PROSITE, 6-fold higher than PF2GO2EC. Overall, this analysis showed a superior performance of our bottom-up method over top-down methods.

Table 4.2 – F1 score and coverage of EnzDP in comparison to top-down methods

		Coverage	
	F1 score	*#EC numbers*	*% predictable ECs*
ProSite	*90.6*	1283	31.1
PF2GO2EC	*79.1*	508	12.3
EnzDP	***97.3***	***3152***	***76.4***
All predictable	-	4126	100

4.3.3 Clustering strategy using domain composition and DEAS

Our clustering strategy is based on a novel idea which explicitly uses domain composition's similarity for subgroup clustering, instead of using sequence similarity. We demonstrate this stringent strategy by comparing it with 2 alternatives (PRIAM and EFICAz), as well as with a BLAST-based EnzDP (which is called EnzDP_bl), on two enzyme families (EC:2.4.1.198 and EC:3.1.3.36) as examples. These two families are chosen randomly with the criteria that they have a reasonable number of enzymes (20-30 members – for illustration purpose), and not too few/too many domains associated (3-6 domains). To make a fair comparison, we used the trained data from Swiss-Prot database (Jan-2011), the same as all other methods. The test data was retrieved from KEGG (Dec-2014).

The first family, EC:2.4.1.198 (Figure 4.5 – *left*), has 27 members (as of Jan-2011 in Swiss-Prot). As can be seen, these members are visually divided into 5 different clusters of the same domain content/architecture. EnzDP clustered this family into 5 subgroups, exactly as intuition. While, PRIAM and EFICAz usually break down EnzDP's subgroups into smaller subgroups, resulted in 7 and 10 subgroups, respectively. This discrepancy comes from the fact that PRIAM and EFICAz use sequence similarity for clustering, while EnzDP uses domain architecture for the similarity metric. If we use BLAST score for clustering (using MCL), the

groups are divided exactly the same way as EFICAz did. In this example, EnzDP gives less number of subgroups than all the others.

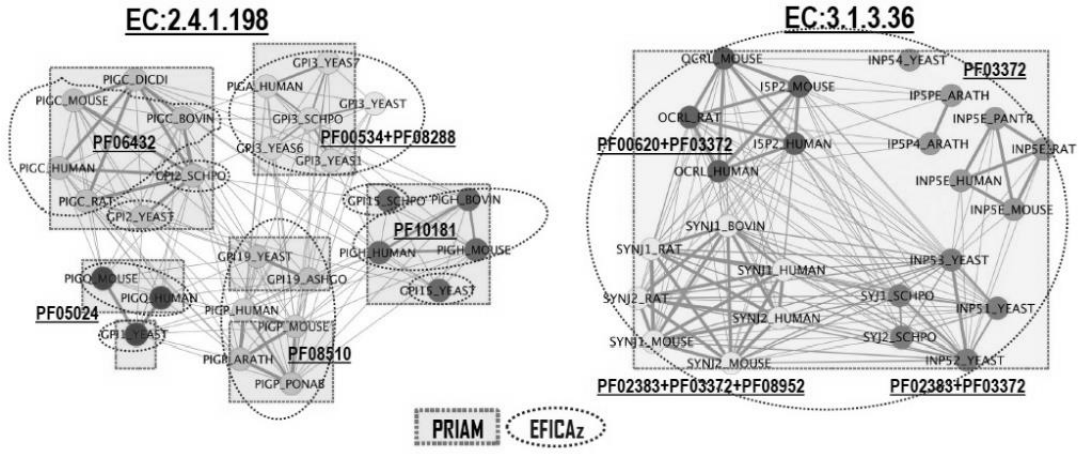

Figure 4.5 – Enzyme sub-family clustering by different methods

Node represents enzyme-protein and its name. Edge's thickness represents sequence similarity between proteins (by Blast e-value). Labels show domain composition of proteins. *Left*: EnzDP clustered EC:2.4.1.198 family into 5 sub-groups, while PRIAM and EFICAz clustered them into 7 and 10 sub-groups, respectively. EnzDP_bl divided this family into 10 sub-groups, exactly the same as EFICAz. *Right*: EnzDP clustered EC:3.1.3.36 into 4 sub-groups, while PRIAM and EFICAz and EnzDP_bl clustered them into 1 group.

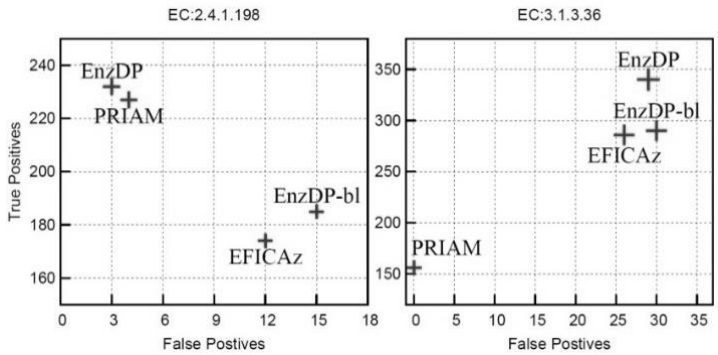

Figure 4.6 – Performance of EnzDP on two datasets, in comparison with other methods

For the second family (EC:3.1.3.36 – 24 members), PRIAM and EFICAz clustered them into one group, while EnzDP split them into 4 subgroups (as shown in Figure 4.5 – *right*). Note that, all members carry the PF03372 domain in common, but different subgroups have different extra domains. In addition, the PF003372 domain is also carried by 13 different enzyme

families (data not shown). Thus, relying on only this domain for clustering and building profile may be misleading. In both examples, EnzDP clustering was more intuitive and reasonable.

Figure 4.6 and Table 4.3 show a comparison of performance between 4 methods. For the first family, EFICAz and EnzDP_bl (used 10 subgroups) suffered both low recall and low precision, compared to PRIAM (7 subgroups) and EnzDP (5 subgroups). While for the second family, both PRIAM and EFICAz (used 1 subgroup) had lower recall than EnzDP (used 4 subgroups). The enforced-version *EnzDP_bl* (using Blast score for clustering), where no DEAS is used, had low recall. EnzDP and EnzDP_bl are only different on the clustering strategy, all other steps were set the same. In fact, *EnzDP_bl* performed as well as EFICAz. In all comparison, EnzDP consistently achieved the best recall and F1 score.

Table 4.3 – Performance comparison among PRIAM, EFICAz, EnzDP_bl, and EnzDP

Method	***#predicted***	***#correct***	***Recall (%)***	***Precision (%)***	***F1 (%)***
		EC:2.4.1.198 (237 sequences from KEGG)			
PRIAM	231	227	95.8	98.3	97.0
EFICAz	186	174	73.4	93.5	82.3
EnzDP	235	232	**97.9**	**98.7**	**98.3**
EnzDP_bl	197	182	76.8	92.4	83.9
		EC:3.1.3.36 (384 sequences from KEGG)			
PRIAM	156	156	40.6	**100**	57.8
EFICAz	312	286	74.5	91.7	82.2
EnzDP	369	340	**88.5**	92.1	**90.3**
EnzDP-1	320	290	75.5	90.6	82.4

Methods were trained on the same Swiss-Prot data, and tested on the same KEGG data. EnzDP-bl is EnzDP with clustering based on Blast e-value (instead of DEAS).

4.3.4 Five-fold cross validation.

In statistical prediction, the following three cross-validation methods are often used to examine a predictor for its effectiveness in practical application: independent dataset test, subsampling test, and jackknife test. However, of the three test methods, the jackknife test is deemed the least arbitrary that can always yield a unique result for a given benchmark dataset. Accordingly, the jackknife test has been increasingly used and widely recognized by investigators to examine

the quality of various predictors (see, e.g. [112-114]). However, to reduce the computational time, we adopted the 5-fold cross-validation in this study, as done by many investigators.

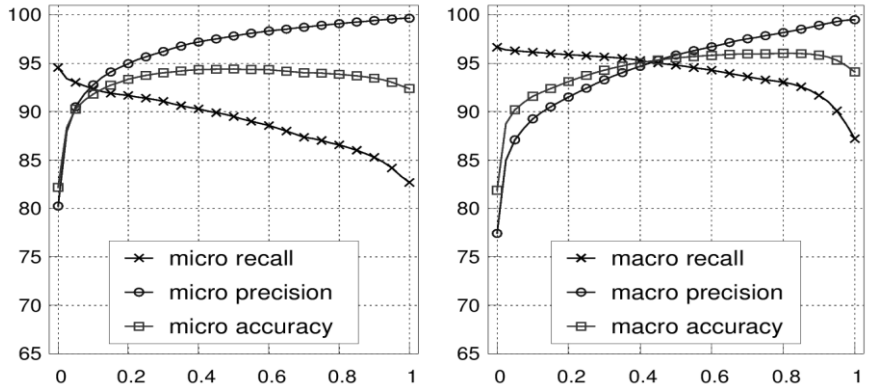

Figure 4.7 – Performance of EnzDP in 5-fold cross validation

Recall, precision and accuracy curves are showing with varying cut-off scores.

In this validation, a *strict* setting was adopted. In training, the method considers false hits are those that are not family members, and considers true hits are those that are family members, but does not recognizes whether a true hit is a true or false sub group member. Thus the threshold for a profile is generous, leaving room for those candidates that are not in same sub group (of the profile), but in same big family (with multiple profiles). However, in testing for each sub group profile, a hit is considered as false hit if it is not correct sub group. Doing so would give more rigorous performance measure. This imitates an extreme situation, when the false hits are similar to the true hits, and thus the number of false positives may increase.

The macro and micro performances were also calculated for the 5-fold cross validation (see Methods). Figure 4.7 shows recall, precision and accuracy curves of EnzDP by varying cut-off threshold. EnzDP achieved micro-recall from 82% to 95%, and micro-precision from 80% to 99%, with the best micro-accuracy of 94.7%. The best macro-accuracy was 96.0%. Macro-recall was better than micro-recall, but, micro-precision was better than macro-precision. In general, the resulted macro-accuracy was always higher than micro-accuracy, for the same cut-off value. This implies that profiles for bigger enzyme families got higher accuracy.

Interestingly, EnzDP got at least 77% macro-precision at zero score threshold, without any cut-off. This indicates that the profiles could reject false members quite accurately. In other words, false members were rarely hit by the profiles, or the hit scores were too small that they were filtered by the preprocessing step in profile searching.

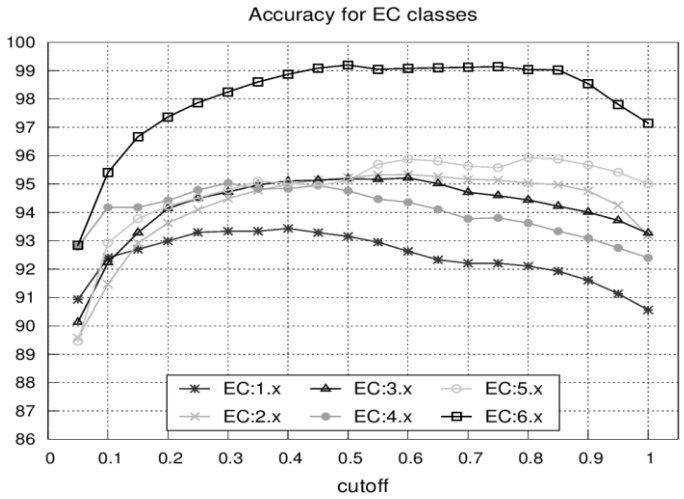

Figure 4.8 – Accuracy for different enzyme classes
Ligases are easier to classify than oxidoreductases.

Figure 4.8 shows the macro-accuracy curves for each enzyme classes. As shown, the EC:6.x class (ligases) got the best accuracy, while EC:1.x class (oxidoreductases) was the worst. This indicates that ligases were easier to classify than oxidoreductases, in general. Shen et al [49] reported an overall success rate of 98.3% for classifying EC:6.x, which is much higher than the success rate for EC:1.x (86.7%). Their success rates for remaining classes (EC:2.x, EC:3.x, EC:4.x, EC:5.x) were in range from 93% to 96% (95.8%, 95.9%, 94.4%, 93.3%, respectively). Our result is mostly consistent with their reported result.

In leave-one-out cross validation, EnzDP was trained with Swiss-Prot data, excluding the enzyme set of a species each time, and then tested on that enzyme set data. This experiment was done in comparison with PRIAM, EFICAz, and ModEnzA on 5 datasets. EnzDP consistently shows improvements on both accuracy and coverage.

4.3.5 Comparison with other methods on recently annotated enzymes.

To directly compare our method with other alternatives, EnzDP was trained on the same datasets as these methods. The testing datasets contain all newly added proteins, exclusively in the Jan-2014 release of Swiss-Prot. We excluded enzymes of new EC numbers, since all methods cannot predict them (see Table 4.1 for details of datasets). Comparison is summarized in Figure 4.9.

Jan11-Jan14 dataset

Figure 4.9 – *left* shows the precision-recall curves of different methods on Jan11-Jan14 dataset. As can be seen, for the same recall, EnzDP got the highest precision among all methods. PRIAM was slightly worse than our method. In fact, EnzDP and PRIAM significantly outperformed the remaining 3 methods. EFICAz achieved a high precision (more than 80%) on average, but its recall was low (less than 37%).

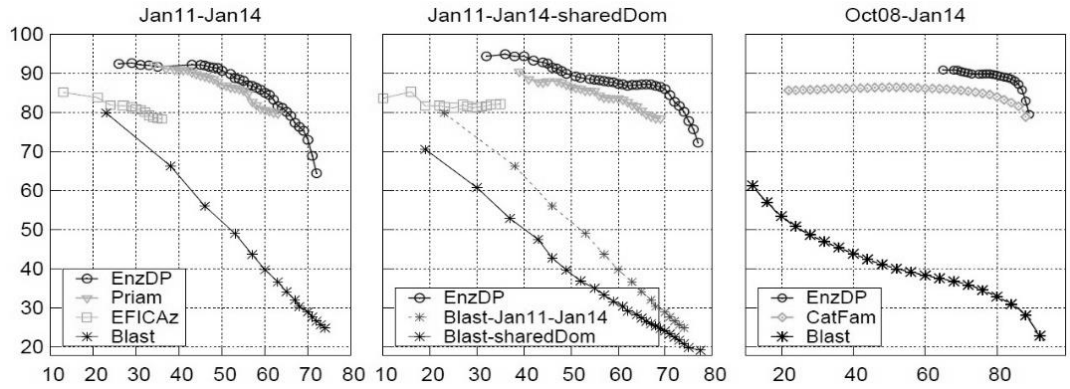

Figure 4.9 – Comparison of Precision-Recall curves on different datasets

On the other hand, BLAST had a base-line performance only. For example, at recall value of 60%, the precision of BLAST is about 40%. At that same recall, PRIAM and EnzDP achieved 2-fold improvement on precision, which was 80% and 85%, respectively. BLAST achieved the highest possible recall, as expected. However, there was a clear trade-off between recall and precision of BLAST, where precision decreased sharply with increasing of recall. At recall of 75%, precision of BLAST went down to 24%.

Oct08-Jan14 dataset

Figure 4.9 – *right* shows the precision-recall curves on Oct08-Jan14 dataset. EnzDP was trained at the same data of Oct-2008 as CatFam. This dataset contains 60024 enzymes, distributed into 1252 EC numbers. EnzDP achieved the best performance among 3 methods. Both CatFam and EnzDP maintained high precisions for this dataset. In fact, their precision slowly decreased as recall increased.

Shared domain dataset

To check the effect of shared domains, we tested methods on the subset of enzymes that carried shared domains. Only domains that are shared between enzymes and at least 10 non-enzymes were considered. This setting makes sure that the domains are not uniquely associated with one enzyme family, and thus they may cause ambiguity. The *subset* of shared domain for Jan11-Jan14 dataset (we called *Jan11-Jan14-shared-domain*) contains 1059 enzymes, associated with 255 EC numbers and formed 1122 EC number enzyme sequence pairs.

Figure 4.9 – *center* shows the precision-recall curves for 4 methods on this dataset (and a projection of BLAST performance on Jan11-Jan14 dataset, for comparison purpose). Performance of all methods had same trend, and EnzDP was the best. Significantly, in comparison with the same curves on the original Jan11-Jan14 dataset, Blast had a clear loss of performance. For examples, at the recall of 60%, precision of BLAST decreased from 40% to 30% (25% loss). At 50% recall, BLAST precision decreased from 52% to 38% (26.9% loss). At the same time, the precision of EnzDP, PRIAM, and EFICAz slightly changed in both directions. In fact, EnzDP, PRIAM had a slightly increased precision.

This analysis shows that EnzDP (as well as PRIAM and EFICAz) was not affected by shared domain issue, while BLAST was. A possible explanation for this failure of BLAST is that, shared domains cause BLAST to match to wrong hits. Thus, if the threshold cut-off for BLAST is not carefully chosen for each family, it may miss true positives and at the same time, give false positives.

4.3.6 Accuracy improved by checking active sites

EnzDP was evaluated with the ability of checking active sites as post-filtering. This version (called *EDP+Site*) accepts a prediction if it satisfies EnzDP filtering threshold, *and* if the input sequence has correct sites matched to active sites of the hit profiles (see Methods). For the dataset in Swiss-Prot release of Jan-2014, there is a small portion of proteins (89311/542258) that have known active site annotated. Thus, we used the *subset* of data that contains active site annotation from Swiss-Prot (we called *AS dataset*), to test EDP+Site. For subset AS of Jan14-Nov11 dataset, 474 enzymes have active site annotation, distributed into 328 EC numbers. For subset AS of Jan14-Oct08 dataset, there are 11576 enzymes with active site annotation, belonging to 538 EC numbers.

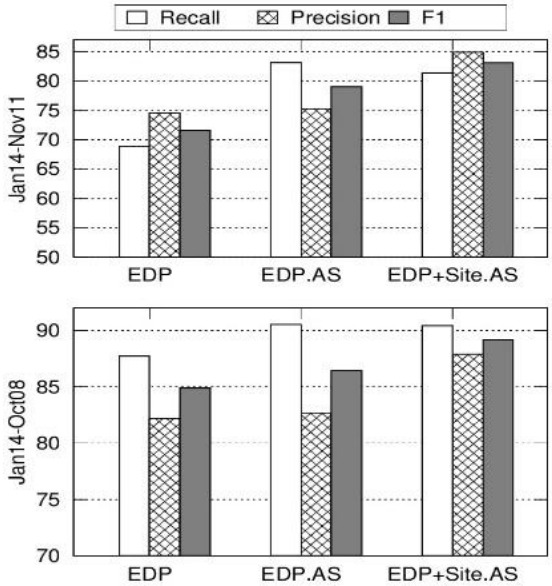

Figure 4.10 – Improved accuracy by checking active sites

EDP denotes overall performance of EnzDP on original dataset. EDP.AS denotes performance of EnzDP on subset (AS) of enzymes that have known active sites. EDP+Site.AS denotes performance on this subset of EnzDP method with stringent active site checking.

Figure 4.10 shows the performance of EDP+Site and two other versions without active site checking. As can be seen, performance of the method (without active site checking) on active-site datasets (EDP.AS) was higher than on the original datasets. Both recall and precision were improved. This indicates that enzyme families with known active-sites were easier to

classify than enzyme families without known active-sites. In other words, active-site information helps improve classification accuracy. EnzDP allows checking of active sites as an option.

EDP+Site (EnzDP with active site checking) had substantial higher precision with slightly lower recall. On the *AS* subset of Jan14-Jan11 dataset, while precision of EDP (without active site checking) was only 75%, precision of EDP+Site was 85%.Thus the precision was significantly improved. The loss of recall was small (from 83% to 81%). For AS subset of Jan14-Oct08, the precision improvement was also significant (from 83% to 88%), and recall loss was negligible. Thus, this analysis confirms that checking active site can improve precision, if active site information is available.

4.3.7 Improvement by threshold setting

To evaluate the improvement of threshold setting, we compared several settings of EnzDP, and compared to a standard BLAST method. Figure 4.11 shows recall-precision curves of the settings on the dataset of *short_adh* (PF00106) domain. This dataset contains 706 proteins, which belong to 90 enzyme families and non-enzymes.

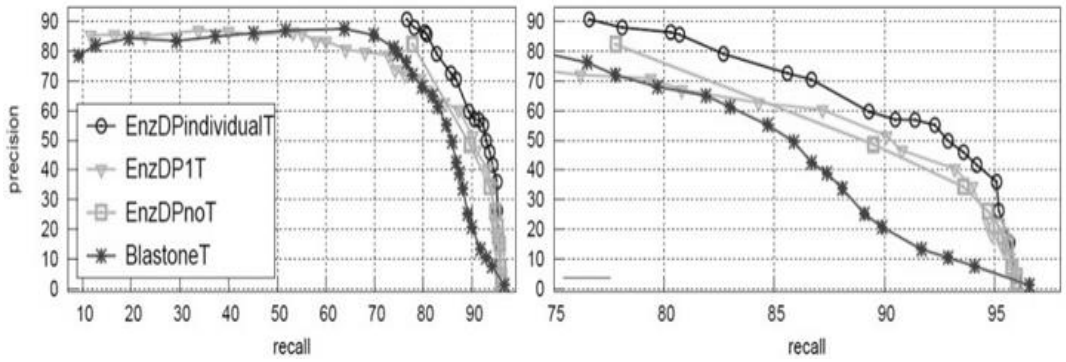

Figure 4.11 – Comparison of different threshold setting strategies on PF00106 dataset
Four settings were included: EnzDP with individual threshold for each profile (EnzDPindividualT), EnzDP with one threshold for all profiles (EnzDP1T), EnzDP with no profile threshold (EnzDPnoT – only ranking of output is used), and BLAST with one common bit-score threshold (BlastoneT). Right chart is a zoomed snapshot of the left chart (from recall 75% onward).

As can be seen, EnzDP with individual threshold had much better precision (for the same recall) compared to other settings. The recall range of EnzDP and EnzDPnoT is also better. For example, these two settings have recall range from 75% to 96%, while for BlastoneT and EnzDP1T the recall range started from a very low 10%, without improvement of precision. In other words, this implies that using individual threshold improves precision greatly, without reducing recall.

4.3.8 Leave one out cross validation comparison

We performed leave one-out cross validation for several genomes, namely *E. coli*, *B. aphidicola*, *P. falciparum*, *H. influenzae*, and *M. genitalium*. Each time a genome was excluded from training, and was used in testing. For each genome, we re-trained EnzDP on data excluding it, and then used it for testing. This imitates the situation of annotating newly sequenced genome. We compared our result with the results of other classifiers that recently developed and achieve high accuracy, namely EFICAz, ModEnzA, and PRIAM. Unfortunately, a direct comparison between all methods was not feasible, since for this validation procedure, each of these methods was trained on data that retrieved at a different timestamp. Thus, we only compared our results with their *reported* results. Comparison is shown in Table 4.4.

Table 4.4 – Performance in leave-one-out cross-validation

	EnzDP	ModEnzA	EFICAz	PRIAM
Training time	*2014*	*2010*	*2004*	*2003*
E. coli Sequences	**93.9/ 92.9**	*92.45/ 84.93*	86.11/ 81.44	-
E. coli ECs	**93.5/ 97.6**	*91.06*/ 88.12	86.26/ 88.87	88/ *92*
B. aphidicola Sequences	**98.3/ 97.9**	*95.63/ 96.33*	93.81/ 94.50	-
B. aphidicola ECs	**98.9/ 98.9**	88.70/ 94.42	*91.53/ 95.37*	91/ 86
P. falciparum Sequences	**61.0/ 82.2**	*60.66/ 75.87*	54.91/ 61.66	-
P. falciparum ECs	61.5/ 75.2	**70.23/ 86.77**	*62.20/ 75.30*	-
H. influenzae Sequences	**97.9**/ *92.3*	-	-	*91*/ **94**
M. genitalium Sequences	**97.9**/ **93.4**	-	-	*87/ 86*

Each time, a genome was excluded from training and was used as testing dataset. The *recall/precision* for each method are shown for enzyme sequence classification (Sequences) and EC number prediction (ECs). The number in **bold** indicates the best result for each row. The statistics of EFICAz, ModEnzA, and PRIAM methods were retrieved from their works.

As can be seen, EnzDP achieved the recalls ranged from 61-98.9% and precisions ranged from 75.2-98.9% for both enzyme sequence and EC number predictions, for multiple genomes. Our result outperformed EFICAz, ModEnzA, and PRIAM for most genomes. In fact, all methods show high accuracy on *B. alphidicola* dataset, and low accuracy on *P. falciparum* dataset. This can be explained by the existence of many other well-annotated *B. alphidicola* strains in the Swiss-Prot. On the other hand, *P. falciparum* has only small number of sequence annotations that were manually reviewed, (157/ 5356) as of Dec-2013.

The statistic in Table 4.4 also shows that, methods at later training time often get higher performance. One of possible reasons is that, later methods had more data to train than previous methods, and thus their performances improved. However, while improvement of ModEnzA (2010) over EFICAz (2004) was not consistent, our method (2014) clearly showed a better performance than all other methods over all datasets (except for *P. falciparum*.)

To test EnzDP as a gap filler on benchmark network datasets described in Chapter 3, section 3.2.3, it was retrofitted and run with default settings. The EC numbers annotations that have 3 or less number of digits were filtered out. The same performance measures were computed. Result is summarized in Table 4.5

Table 4.5 – Performance of EnzDP on 5 networks

	iIN800	*iWV1314*	*iIB711*	*iHD666*	*iMA871*
Precision	85.2	53.1	56.8	59.4	49.7
Recall	87.3	64.3	65.1	66.4	57.2
F2	86.9	61.7	63.3	64.9	55.5

In comparison with other gap filling methods (PFP-GF, EFICAz-GF, B2G-GF) showing in section 3.3.4, EnzDP got the better recalls and higher F2 scores. However, its performance is lower than that of MeGaFiller (except on iIN800 dataset, where EnzDP is better than MeGaFiller). This may suggest that using EnzDP for gap filling is also possible and effective. In fact, EnzDP has high accuracy and high coverage. Thus, it was able to propose more correct candidate for gap filling. The result is consistent across all the five datasets.

4.4 Conclusions

Determining the entire complement of enzymes and their enzymatic functions is a fundamental step for reconstructing the metabolic network of cells. High quality enzyme annotation helps in enhancing metabolic networks reconstructed from the genome, especially by reducing gaps and increasing the enzyme coverage. Currently, structure-based and network-based approaches can only cover a limited number of enzyme families, and the accuracy of homology-based approaches can be further improved. Bottom-up homology-based approach improves the coverage by rebuilding Hidden Markov Model (HMM) profiles for all known enzymes. However, its clustering procedure firmly relies on BLAST similarity score, ignoring protein domains/patterns, and is sensitive to changes in cut-off thresholds.

Here, we use functional domain architecture to score the association between domain families and enzyme families (DEAS). The DEAS score is used to calculate the similarity between proteins, which is then used in clustering procedure, instead of using sequence similarity score. We improve the enzyme annotation protocol using a stringent classification procedure, and by choosing optimal threshold settings and checking for active sites.

Our analysis shows that stringent protocol EnzDP can cover up to 90% of enzyme families available in Swiss-Prot. It achieves a high accuracy of 94.5% based on 5-fold cross validation. EnzDP outperforms existing methods across testing scenarios. Thus, EnzDP serves as a reliable automated tool for enzyme annotation and network reconstruction.

Chapter 5

NetA: assembling metabolic network from genome annotation

5.1 Background

5.1.1 Overview

The problem of metabolic reconstruction is well established, yet remains a difficult problem [13, 23]. It may take years to complete and requires a lot of manual labour of human experts, especially manual quality control is unavoidable. Therefore, it is sensible to develop automated tools in order to speed up reconstruction process and reduce human involvement. Indeed, the reconstruction pipeline (see Figure 2.4, section 2.1.2) gives a basic guideline to divide the whole process into smaller tasks; each task can be resolved separately using different tools. This chapter addresses the *network assembly* problem – the central part of the pipeline.

In literature network assembly is referred to as network reconstruction problem. In fact, this network assembly problem may show that our metabolic knowledge gained from all previous steps is not complete. Especially, it reveals the metabolic/reaction gaps that previous steps failed to show. In consequence, gap filling step will need to be carried out, and the whole reconstruction process may need to be repeated from the first step.

Therefore, the most important issue in the assembly problem is to achieve a high quality network, which can be evaluated by the network connectivity, the reaction's coverage, and the amount of missing knowledge. The connectivity shows that all the metabolic reactions are feasible in the network, without any dead-end. With connectivity, the network is usable for modelling and simulation in later steps. The reaction coverage is the most important factor of

high quality. It is counted for only those reactions with enzyme-gene evidence. Thus, quality of assembled network is strongly affected by quality of enzyme annotation in previous step. In addition, assembly method should not miss any known reactions. Last but not least, it is desired that the network contains the minimum missing metabolic knowledge, with minimum number of metabolic gaps.

Another important consideration in network assembly is the possibility of ***quickly reproducing*** the network if any new data is available. This is usually the case when gaps are filled or new knowledge is gained from later steps, such as, gap filling, model verification and simulation. Therefore, ***automated tools*** for minimizing manual work are desired.

5.1.2 Pros and cons of current methods

There are two approaches to network assembly problem, namely pathway-based and network-based. The first approach firmly relies on reference pathways database – such as KEGG [31] and BioCyc [41], while the second approach relies on reconstructed networks of closely related species, such as *Vongsangnak et al.* [3]. Chapter 2, section 2.2.2 gives more details on these approaches. Here we discuss the pros and cons of each approach.

Pathway-based methods cannot reconstruct new pathways, since it follows predefined templates that do not allow to add new reactions. However, since biological function organized by pathways, it makes sense to look at pathway individually. Intuitively, it is relatively easier to handle pathways separately than handling whole network at once. Furthermore, this approach is reliable, as known metabolic knowledge has been summarized in reference pathways of KEGG.

Network-based methods can predict novel pathways, since new reactions can be added into the network, which is useful for predicting new network from scratch. It reconstructs a whole network at a time. However, network-based methods may require the constrained conditions and/or experimental data, which may not be available from the beginning. The main concern about network-based approach is to fill reaction gaps, but it leaves those filled reaction

gaps as metabolic gaps. In fact, both approaches still allow gaps in the resulting network, and the added reactions may have no gene evidence.

Essentially, filling network gaps by *adding new reactions* may introduce new network gaps. In addition, adding new reactions increases the problem size. Thus, doing so may further complicate the problem. In our approach, we use reference pathways as the templates to start with.

In contrast to adding new reactions as in network-based approach, we *delete un-necessary reactions* instead. Given the set of known enzymes and a template pathway, we would divide the reactions in the template into 3 categories: known (or necessary) reactions, corresponding to the known enzymes; unknown but necessary reactions (those that do not have known enzymes associated, but are necessary for the connectivity of the pathway/network); and, unknown and un-necessary reactions (those that do not have known enzymes as well as are not critical to the pathway's connectivity). Unfortunately, the 2 last categories are interdependent and are not easy to divide. This is because the connectivity can be achieved with many different subsets of those unknown reactions, while it's not obvious which subset should be chosen. Our proposed algorithm tries to predict the best subset of unknown reactions to keep, and deletes the remaining from the template pathway. An advantage of the deleting reaction approach is that, it guarantees no network gaps by always maintaining the connectivity of the network during deletion.

5.1.3 Our approach

We aim to develop computational methods for automating the process of building a backbone network, from genome annotation. We try to improve network quality, focusing on network's connectivity with high enzyme coverage and minimum metabolic gaps. Our methods do not use experimental data, and thus, the prediction result requires further experimental verification. Validation requirements are out of current research's scope.

Finding that the quality of network reconstruction strongly depends on quality of annotation set, we first focus on improving the enzyme annotation set by using multiple sources

of annotation, including computational predictions by EnzDP and MeGaFiller. The initial reaction set is considered reliable with gene evidence.

We follow the *pathway-based approach*, in which we use KEGG reference pathways to predict the presence of all possible metabolic pathways for the given set of enzymes. We then optimize each predicted pathway to get minimum number of gaps. Finally we merge those optimized pathways into a network, via combining resultant KGML files.

For predicting presence of pathways, we assume that a pathway is present in a genome *G*, if majority of its reactions have enzymes expressed in *G*. Thus, our approach relies on a presence ratio, which is the ratio of enzyme-supported reactions over the total number of reactions included in the reference pathway. The threshold of presence ratio will be tuned by experimental training datasets.

The next step after predicting presence of pathways is pathway optimization. Our approach uses the reference template to optimize the pathway for the given set of enzyme annotation. Specifically, for each reference pathway *P* that is present in the genome of interest, we try to find a minimum sub-pathway *P**, such that *P** is connected, and contains all possible known reactions for the genome of interest (in that pathway) and minimum number of missing enzyme reactions. We prove that the problem of finding optimal pathway is NP-Hard. Consequently, an approximation algorithm called MiniPath is developed.

After that, a network is assembled by merging these optimized pathways. This can be done by merging the resulting KGML files of optimized pathways, using available tool KGML2SBMLconverter. And finally, with the use of the same tool, SBML format of the reconstructed network will be produced. More details are presented in the following section.

5.2 Methods

5.2.1 General description and definitions

First, general definitions and descriptions on network assembly problem will be presented. Then the algorithms for predicting pathway presence, finding optimal pathways, and network assembling will be described in details.

Reaction: a metabolic reaction *r* transforms a set *S* of substrate metabolites into a set *T* of product metabolites. The set *M* of metabolites of reaction *r* is the union set of *S* and *T*. Example of KEGG R00303 reaction (Fig 1.1 - bottom):

$$r: C00092 + C00001 = C00031 + C00009$$

$$S_r = (C00092, C00001)$$

$$T_r = (C00031, C00009)$$

$$M_r = S_r \cup T_r = (C00001, C00009, C00031, C00092)$$

Universal set R of reactions: set of all known metabolic reactions. This set is the global metabolic network, which can be retrieved from reaction databases.

Pathway: a pathway *P* is a set of some reactions:

$$P = (r_i \,|r_i \in R)$$

Note that, by the above definition, pathway may or may not be connected. To describe connectivity of pathways, the notion of ***reachability*** is used, as the following.

Reachability

Given a pathway *P* and a set *S* of source metabolites, a *metabolite m is reachable* in a pathway *P* from *S*, denote by relation *reachable (m|P,S)* if:

- There is a reaction *r* in *P* such that *r* directly transforms some metabolites in *S* to *m*, that is: $S_r \subseteq S$ and $m \in T_r$
- Or, *m* is one of product metabolites of a reachable reaction r_a: $m \in T_{r_a} |\ reachable(r_a|P,S)$

A reaction *r is reachable in P from S,* denote by relation *reachable (r|P,S)*, if *r* is a reaction of *P*, and either:

- r is directly reachable from S, that is, $S_r \subseteq S$, or
- All substrate metabolite m of r is reachable: $\forall m \in S_r, reachable(m|P,S)$

Pathway's outcome: outcome of a pathway P given a set S of source metabolites, $O(P \mid S)$, is set of all metabolites that are reachable in P from S, that is,

$$O(P|S) = \bigcup_{reachalbe(r|P,S)} T_r$$

In other words, it is the union of all product metabolites of reachable reactions in P from S.

Pathway completeness: The completeness of a pathway P towards a set T of required target metabolites given set S of source metabolites is defined as:

$$C(P|S,T) = \frac{|O(P|S) \cap T|}{|T|}$$

In other words, it is the ratio of target metabolites that reachable from S, to the target set T. This ratio is easy to compute, using graph traversal algorithm like BFS to check the reachability for each metabolite and reaction.

5.2.2 Find pathway problem

Given a universal set R of reactions, set S of source metabolites, and set T of target metabolites, and a pathway completeness threshold w, find a pathway $P \subset R$ such that:

$$C(P|S,T) \geq w$$

Note that, if we set $w = 1$, we require that all target metabolites are reachable, or in other words, the pathway must be connected.

Find pathway with known reaction set

Given a universal set R of reactions, a set of initially known reaction R_0, set S of source metabolites, a set T of target metabolites, and a pathway completeness threshold w, find a pathway $P \subset R$ such that $R_0 \subseteq P$ and:

$$C(P|S,T) \geq w$$

In fact, this problem is the same as the Find pathway problem, by encoding the requirement of R_0 into the set of target metabolites. The fact that R_0 is known to be occurred in P is equivalent to that *all metabolites in R_0 are reachable* in P.

With the setting $T' = \bigcup_{r \in R_0} M_r \cup T$, the problem becomes finding P such that $C(P|S,T') \geq w$

Find minimum pathway problem

(Find-Minimum-Pathway) Given a universal set R of reactions, set S of source metabolites, a set T of target metabolites, and the pathway completeness threshold w, among all pathways P ⊂ R such that:

$$C(P|S,T) \geq w$$

find the pathway P with minimum size: |P*| ≤ |P| for all P.*

This problem is an NP-Hard problem. In a setting up to atom level metabolic network, Pitkanen et al [83] have proved its NP-Hardness. The same idea can be applied for metabolite level. For illustration purpose, the idea will be adapted and presented next.

5.2.3 Find-Minimum-Pathway is NP-Hard

Proof:

NP-Hardness of Find-Minimum-Pathway can be shown via a reduction from the Minimum-Set-Cover problem. In a nutshell, we will show that:

Minimum-Set-Cover $\leq_p$ *Find-Minimum-Pathway*

Minimum-Set-Cover: *Given a universe U set of n elements U = (1,2,...,n), and collection C of k sub sets whose union equals the universe, C = (C_1, C_2, ...,C_k),* $\bigcup C_i = U$*. Find the smallest sub collection C* of C, such that union of C* equals the universe, that is* $\bigcup_{C_i \in C^*} C_i = U$ *,* $|C^*| \rightarrow min$*.*

Given an instance *(U, C)* of the Minimum-Set-Cover, an instance of Find-Minimum-Pathway, *(R, S, T, w)*, can be reconstructed as following.

For each element *u* of *U*, introduce 2 metabolites s_u and t_u. Set $S = \bigcup s_u$ and $T = \bigcup t_u$. For each sub set C_i of *C*, introduce a reaction r_i. Suppose C_i = *(j,k,l,...)*, then correspondingly, the substrate metabolites for r_i are: s_j, s_k, s_l, ..., and the product metabolites for r_i are: t_j, t_k, t_l, ...

Set $R = \bigcup r_i$ and *w = 1*. Note that, *w = 1* means Find-Minimum-Pathway requires finding a complete pathway, i.e., all metabolites in *T* are reachable.

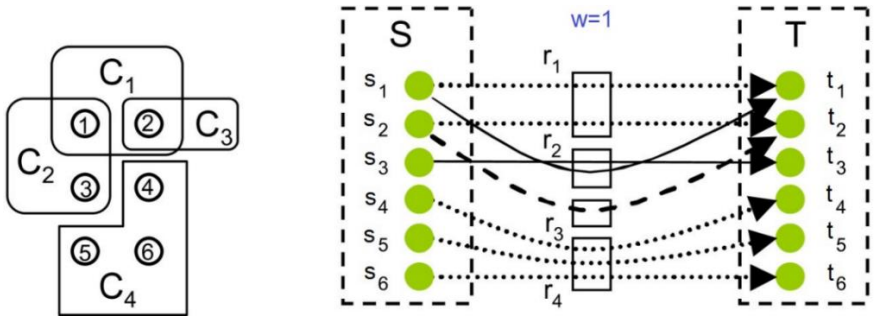

Figure 5.1 – Reduction of Minimum-Set-Cover to Find-Minimum-Pathway

Picture adapted from Pitkanen et al. **[83]** ***Left***: An instance of Minimum-Set-Cover with $U = (1,2,3,4,5,6)$; $C = (C_1,C_2,C_3,C_4)$, $C_1 = (1,2)$, $C_2 = (1,3)$, $C_3 = (2)$, $C_4 = (4, 5, 6)$. ***Right***: Find-Minimum-Pathway instance with 12 metabolites and 4 reactions. Pathway completeness $w = 1$. Source set of metabolites $S = (s_1,\ldots,s_6)$, target set of metabolite $T = (t_1,\ldots,t_6)$. Arrows denote mapping of metabolites over reactions. In particular, mappings shown with similarly dashed arrow lines belong to the same reaction.

Assume P is a solution of Find-Pathway problem. Clearly, the corresponding solution for Set-Cover problem is: $C_p = (C_i | \, r_i \in P)$. And on the other hand, it can be seen that, if a solution of Set-Cover problem is C_p, then the corresponding to the solution for Find-Path problem is P. Thus, an optimal solution C^* can also be reconstructed from the optimal solution P^*.

It is not hard to see that, all the above reconstruction can be done in polynomial time. Thus, Minimum-Set-Cover is polynomial time reducible to Find-Minimum-Pathway. The NP-Hardness of the former implies the NP-Hardness of latter. **Q.E.D**.

5.2.4 Network assembly problem:

Metabolic network: Metabolic network N is a collection of pathways:

$$N = (P_i)$$

Metabolic network assembly problem: Given a set of known reactions N_0, reconstruct a metabolic network N that contains N_0, i.e. for any r in N_0, there exists P in N such that r is in P

Predicting presence of pathways

To reconstruct a metabolic network from a set N_0 of known reactions, we use KEGG pathway templates, and reconstruct pathway by pathway. So, the first task is to decide whether

a template pathway should be included into the result network. Intuitively, if a majority of reactions in a reference pathway *P* is known, then *P* should be included. To qualify the presence of pathway P, we define the presence ratio h_P as the following.

$$R_0 = P \cap N_0$$

$$h_P = \frac{|R_0|}{|P|}$$

If the presence ratio h_P is higher than a predefined threshold, *P* will be included into the list of pathways for the result network. The pathway *P* is then further optimized to get a minimum pathway before it can be added into the final network.

Finding minimum pathways

Given a reference pathway *P*, and a set of known reactions R_0 in *P*, we want to find a minimum pathway *P** that covers R_0, is connected, and contains minimum number of gaps. As discussed previously, this problem is the same as finding minimum pathway, with an extension to the set of target metabolites $T' = \bigcup_{r \in R_0} M_r \cup T$

Algorithm *MiniPath* *(P, Sp, Tp, R_0):*

Input:
P – A reference pathway, with input and output metabolites *Sp* and *Tp*.
R_0 – A set of known reactions
Begin
  Set $T = \bigcup_{r \in R_0} M_r \cup Tp$
  Set *P' = P*
  Set *D = P – R_0*
  for each reaction *d* in *D*:
    P' = P' – (d)
    check if pathway is complete, i.e. *C(P'| Sp, T)* >= 1
      if not: *P' = P' + (d)*
  return *P'*
End
Output:
P' – A minimal pathway

Due to the NP-Hardness of Find-Minimum-Pathway problem, we propose a simple approximation algorithm *MiniPath* to find a minimal pathway. We assume each of KEGG reference pathways is complete. That is, all reactions in a reference pathway are reachable from

pathway's input. This is reasonable assumption, since KEGG pathways are manually curated and drawn. Our approximation idea is to delete unnecessary reactions from the template, while keeping the reachability of all required reactions (in R_0). The reachability can be easily checked using a graph search, e.g. BFS algorithm.

In a nutshell, *MiniPath* first computes the set *D* of reactions that might be deleted from *P*, $D = P - R_0$. Then it tries to delete each reaction *d* of *D* from the result pathway: $P' = P' - (d)$. If after the deletion, not all required metabolites are reachable, then it reverts the deletion, $P' = P' + (d)$. In fact, MiniPath will stop only if none of remaining reactions can be deleted without violating the pathway connectivity. Therefore, it finds a local optimal solution.

Complexity of MiniPath algorithm

MiniPath runs in quadratic time and takes linear space. Indeed, checking reachability for all reactions and metabolites can be done by any graph traversal algorithm, such as BFS or DFS, and it takes linear time. Thus, overall MiniPath runs in quadratic time, that is, $O(|P|^2)$. It uses no more than $O(|P|)$ memory.

Approximation ratio of MiniPath

In the worst case, MiniPath deletes only 1 reaction, while the optimal solution can delete all the other |*D*|-1 reactions. In this case, $|P'| = |P| - 1$, while for the optimal solution $|P^*| = |R_0|+1$

We have, the worst case approximation ratio *r* is:

$$r = \frac{|P'|}{|P^*|} = \frac{|P| - 1}{|R_0| + 1} < \frac{|P|}{|R_0|} = \frac{1}{h_p}$$

where, h_p is indeed the presence ratio of the pathway *P*. So, it means that if we set the presence ratio threshold to be ½, then the *MiniPath* has an approximation ratio of 2.

Network assembly algorithm

Given an initial reaction set, the final goal is to build a metabolic network that contains these reactions. Following the pathway-based approach, our method first predicts the presence of pathways for those initial reactions, using KEGG pathway templates. Those pathways with presence ratio greater than a predefined threshold h_0 will be included into the network. Each of

these pathways will be optimized by MiniPath algorithm to get a minimal pathway. At the end, these optimized pathways will be merged into the final network. Our network assembly algorithm ***NetA*** is following.

Algorithm *NetA (N0, K, h_0)*

Input:
  N_0 – Initial reaction set
  K – KEGG reference pathways
  h_0 – pathway presence threshold
Begin
  $N = ()$
  For each pathway P of K:
    $R_0 = P \cap N_0$
    Calculate $h_P = |R_0|/|P|$
    If $h_p >= h_0$:
      $P' = MiniPath(P, Sp, Tp, R_0)$
      $Merge(N,P')$
  Return N
End
Output
  N – metabolic network

Procedure *Merge(N,P')* is in fact merging the pathway *P'* into a network *N*. It first uses the KGML file retrieved for the reference pathway *P*, and modifies it according to reactions in *P'*. It then merges the modified KGML file into the KGML file of the resulting network. Finally, it uses an available tool called KGML2SBMLConverter to convert the merged KGML file into a network of SBML format. The merging step runs in linear time.

Since *MiniPath* runs in $O(n^2)$ time, where n is the maximum number of reactions in a reference pathway, it implies that *NetA* algorithm runs in $O(k.n^2)$ time, where k is the number of reference pathways available.

5.2.5 Experimental setting

To tune for the optimal presence ratio threshold, the sets of manually annotated pathways of *S.cerevisiae* (72 pathways) and *E.coli* (82 pathways) retrieved from KEGG are used as a benchmark. NetA predicts pathways for those genomes by varying the presence ratio

thresholds. For each value of presence ratio, sets of predicted pathways by NetA are compared with the benchmark. Ratio that produces the best Jaccard Score is chosen as threshold for NetA.

After tuning for optimal presence ratio, NetA is used to find optimal pathways. For each pathway, result optimized by NetA is compared to the benchmarks. The set of gene-reaction relationships in resulted optimal pathway is compared to that in the benchmark, by computing the *recall* and *precision* values:

$$Recall = TP / (TP + FN)$$

$$Precision = TP/(TP + FP)$$

TP is number of gene-reaction pairs that predicted and correct compared to the benchmark. FP is number of pairs that predicted but not correct. FN is number of pairs that are in the benchmark but not predicted.

5.3 Results

5.3.1 Presence ratio

NetA depends on a presence ratio to predict the set of pathways that should be included into the final network. To obtain an optimal value of ratio threshold, parameter tuning was performed as following. For each value of ratio threshold *r* from 0.05 to 0.75 with step of 0.01, NetA uses *r* to predict pathways for *S.cerevisiae* and *E.coli*. The result is then compared with benchmark data, and a Jaccard score is computed. Best ratio was deemed to be chosen at the best Jaccard score. To get a better sense about the dependency of NetA on the input initial annotation sets, multiple annotation sets were used. These annotation sets were made by EnzDP with different cut-off likelihoods.

Figure 5.2 shows plots of Jaccard scores against presence ratio values, for multiple input initial annotation data of *S. cerevisiae* and *E. coli*. Optimal values were obtained in range [0.17, 0.25], with best value around 0.2 point. As can be seen, this value is consistent for both SC and EC, across multiple annotation inputs.

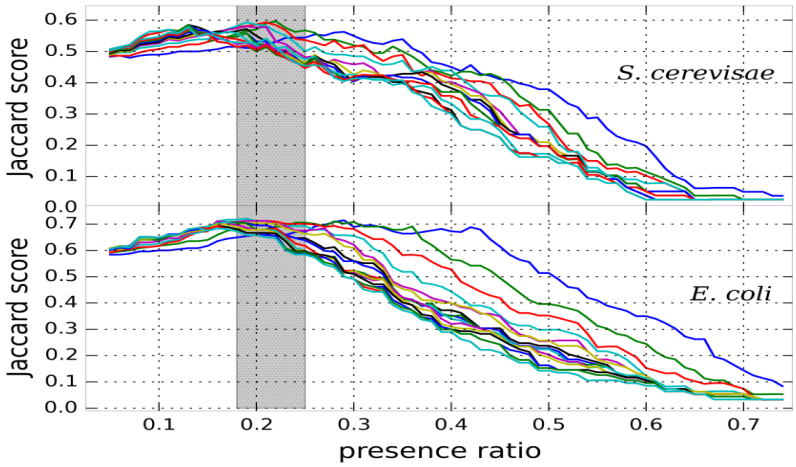

Figure 5.2 – Tuning pathway presence ratio

Jaccard scores were plotted again each presence ratio value in range [0.05, 0.75] with step of 0.01. Optimal value was obtained around the ratio value *r = 0.2*.

A small value of optimal presence ratio is expected, since a specific genome may have a small number of metabolic reactions expressed for a specific biological pathway. In fact, a reference pathway contains all possible reactions for all known species across all 3 domains of life. For majority of pathways, a specific genome of interest may have a small portion in common with the reference pathway. The optimal value of 0.2 achieved in our experiments suggests that, an organism specific pathway only covers about 20% of reactions in the reference pathway. It also showed that, with increasing this ratio, the corresponding Jaccard score is sharply decreasing. Thus, we recommend a ratio in range of [0.17, 0.25] for practical applications.

5.3.2 Pathway optimization

After predicting the set of pathways that are present in the genome of interest, NetA uses the *MiniPath* algorithm (see Methods for more details) to optimize predicted pathways. The overall idea is to delete those unnecessary reactions from the reference pathways, while maintaining the reachability of all metabolites in the targeted set. MiniPath does not guarantee a global optimum solution, however, it will find a local minimal pathway, in which deleting any more reaction will disconnect the current pathway.

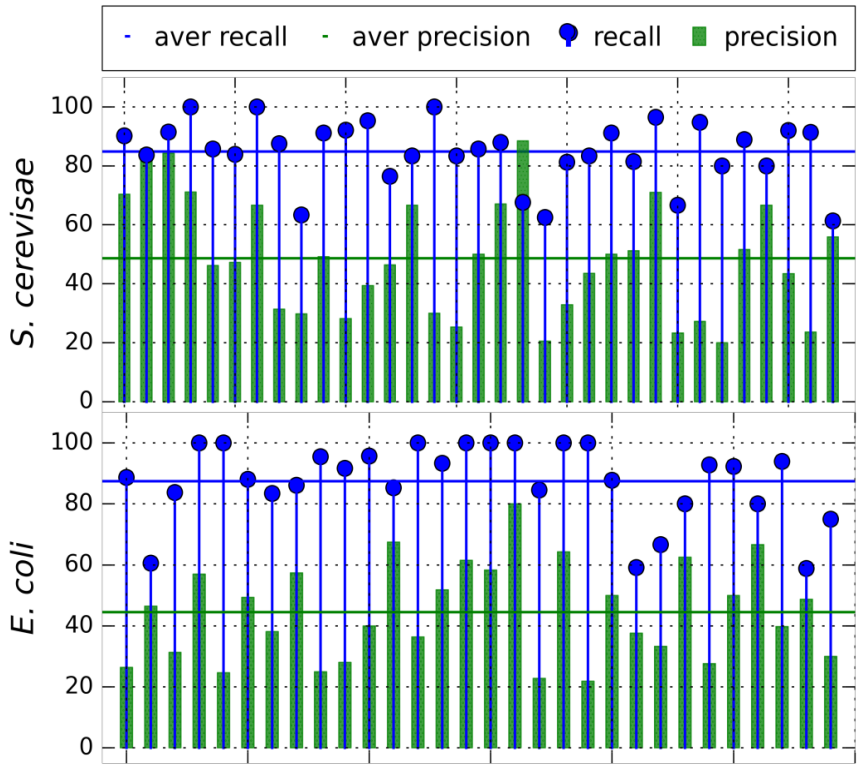

Figure 5.3 – Performance of NetA for optimizing pathways in *S. cerevisiae* and *E. coli*

To evaluate the effectiveness of MiniPath algorithm, NetA was run on *S.cerevisiae* and *E.coli* benchmark datasets. For each pathway the predicted set of reactions was compared to the benchmarks. Performance on common pathways was summarized in Figure 5.3. As can be seen, MiniPath achieved an average *recall* of 84.8% (87.4%) and an average *precision* of 48.6% (44.5%) on *S.cerevisiae* (*E. coli*) dataset. This suggests that about 85% of reactions in the benchmark is agreed with MiniPath. While, almost half of the predicted reactions is agreed with the benchmark.

The fact that NetA achieves a high coverage, but relatively low precision suggests that, NetA predicts more reactions that are not in the current benchmark. Provided that the benchmark itself is not fully complete, the low precision of NetA is expected.

5.4 Discussion and conclusion

In this work, we described the network assembly problem and showed that it is an NP-Hard problem. We then developed an approximation algorithm NetA for assembling a genome-scale metabolic network from a given set of enzyme annotation. First, NetA uses KEGG reference templates to predict which pathways are present in the genome. Analysis showed that, a reference template should be included into the network if it has at least 20% known reactions with annotated enzymes. Then, NetA uses MiniPath algorithm to find a minimal pathway from the given template. MiniPath is an approximation algorithm which tries to delete a maximum number of missing gene reactions from the template while maintaining the pathway connectivity. Finally, NetA merges those optimized pathways into a network using the KGML2SBMLconverter tool.

Experiments on *S. cerevisiae* and *E. coli* dataset showed that MiniPath algorithm can cover 85% of the benchmark reactions over all predicted pathways, with an average precision of 45%. In fact, it predicts more reactions than the given dataset, and many of those reactions have support annotations, suggested by EnzDP.

Although MiniPath algorithm can quickly produce a minimal sub-pathway, NetA overall logic has several ***limitations***. Firstly, NetA assumes that, a reference pathway represents a functional unit that all of its reactions are relevant. However, a template pathway may be much broader than the actual biological pathway in the specific organism. Analysis shows that, only a small portion of reference pathway is present in any specific organism. In other words, a reference pathway is the superset of organism pathways, across all life domains. Thus, it would make more sense if we consider those pathway branches of the relevant organisms only.

Secondly, NetA assumes existence of all input and output metabolites, deriving from the reference pathway. These metabolites depend on the pathway's boundary, but in reality, the pathway's boundary is not well defined. In other words, not all these metabolites necessarily exist in the target organism. Knowing the existence/non-existence of metabolites would help NetA narrowing down the set of constraints to optimize.

Finally, NetA does not leverage on the importance of different reactions. In reality, pathways are interconnected and overlapped, thus there are many reactions that belong to different pathways. These reactions, in many cases, are more important than the others. Thus they should have more weight to keep in the reaction deletion process. Currently, NetA consider all unknown reactions with equal weight.

In ***conclusions***, NetA is an algorithm to quickly predict a minimum connected network for an organism. It works by deleting un-necessary reactions from the reference pathways that are not found in the organism while ensuring connectivity of the reduced network. Although the current logic of NetA is simple, it is an important step in the network reconstruction pipeline that ties together with EnzDP and MeGaFiller. In this pipeline, EnzDP is first used to annotate enzymes given the genome data, then NetA identifies the pathways and build a minimal connected network, and finally, MeGaFiller fills the metabolic gaps in that initial network. If any additional data is available, the pipeline can be repeated with any or all of these steps. This pipeline becomes useful tool for quickly reconstructing a metabolic network from genome.

Chapter 6

Application of NetA and the reconstruction pipeline

6.1 Introduction

This chapter presents an application of NetA and the reconstruction pipeline on reconstructing a draft network for a fungus *A. oryzae*, using EnzDP, NetA, and MeGaFiller. This fungus plays an important role in food production industry [20, 115]; building its metabolic model is essential to in-silico analysing metabolic features. The first and only genome-scale metabolic network of this fungus was reconstructed in 2008, by *Vongsangnak et al* [3]. Our result suggests that the current network can be expanded about half in enzyme gene coverage.

6.2 Methods

Genome data of A. oryzae was retrieved from DOGAN database. The annotations were ignored and only protein sequences were used in this work. First, EnzDP with default settings was run on genome sequences of *A. oryzae* to annotate enzymatic functions. Annotation was validated if the assignment between genes and enzyme families have any support domain family annotated by Pfam.

After that, NetA algorithm was run on *A. oryzae* dataset with the annotation made by EnzDP. The presence ratio was set at 20% as default. After predicting pathways' presence, MiniPath was run to optimize those template pathways. Optimized pathways were then merged

into a back-bone network of metabolic reactions. The network in .kgml format was finally converted into .sbml format.

After getting NetA result, MeGaFiller with common-2 settings (see section 3.3.5 for more details) was applied on the network produced by NetA. We use common-2 settings assuming no prior network has been available. This serves the purpose of reconstructing network from scratch. From that network, the list of metabolic gaps were extracted and used as input for MeGaFiller. The predicted candidates were then mapped back to the network.

6.3 Results

6.3.1 Enzyme annotation by EnzDP

In the first step, EnzDP was run to predict enzymes for *A. oryzae* genome. Table 6.1 shows enzyme content predicted for *A. oryzae*. EnzDP found 1923 genes have enzymatic functions, across 962 enzyme families (EC numbers). In result, EnzDP assigned 2210 pairs between EC-numbers and genes, in which 1571 (81%) of the pairs have support Pfam evidence.

Table 6.1 – Enzyme annotation made by EnzDP on *A. oryzae* genome

	EC numbers	*Enzyme genes*	*EC-gene pairs*	*Pairs with Pfam support*
EnzDP	962	1923	2210	1571 (71%)
iWV1314	651	1246	1375	918 (67%)
Shared count	460	843	719	586 (82%)
Shared coverage	72.2 %	67.7 %	52.3 %	-

Shared count is the number shared between EnzDP predictions and *iWV1314* network. Shared coverage is the percentage of shared count over the *iWV1314* network.

In comparison with annotation included in [3], EnzDP shared 460 unique EC numbers, which is 72.2% of the number of unique EC numbers in *iWV1314* network. The number of genes (pairs) in common is 843 (719, respectively). Significantly, EnzDP predicted additional 51% enzyme families and 54% of novel enzyme genes that missed by the *iWV1314*. This represents a potential expansion for *A.oryzae* network.

6.3.2 Network assembly by NetA

NetA predicted 119 metabolic pathways for *A. oryzae*. On average, each pathway after optimization contains 23 reactions. The average enzyme coverage in all pathways is about 72.4%. Table 6.2 shows top 15 dense pathways with highest coverage.

It can be seen from the table that there still has a significant number of metabolic gaps. The average percentage of reactions without enzymes is about 27.6%. This may indicate that MiniPath algorithm is not optimal, as it included those metabolic gaps reactions to make the network connected. It also can be explained by the fact that the initial set of reactions is not complete, thus allowing holes in the final network. Anyway, the result produced by NetA revealed that our current metabolic knowledge on *A. oryzae* is far from complete.

Table 6.2 – Top 15 dense pathways with highest number of reactions

pathway **id**	**pathway name**	Reaction count	Gene count	#Reactions without genes	Enzyme coverage
00685	Pyrimidine metabolism	87	75	16	81.6
00676	Glycerophospholipid metabolism	64	85	17	73.4
00675	Cysteine and methionine metabolism	62	79	14	77.4
00673	Glycine, serine and threonine metabolism	59	115	15	74.6
00686	Propanoate metabolism	48	113	14	70.8
00629	Chlorocyclohexane and chlorobenzene degradation	47	64	7	85.1
00621	Galactose metabolism	45	45	9	80.0
00620	Pyruvate metabolism	43	77	8	81.4
00622	Valine, leucine and isoleucine degradation	42	63	8	80.9
00690	Alanine, aspartate and glutamate metabolism	39	37	11	71.8
00632	Fatty acid degradation	37	45	9	75.7
00628	Steroid biosynthesis	37	33	6	83.8
00691	Glycolysis / Gluconeogenesis	35	73	9	74.3
00696	Sphingolipid metabolism	34	73	9	73.5
00624	Lysine biosynthesis	31	48	5	83.9

Note: Enzyme coverage (last column) is counted as percentage of number of reactions that have gene annotation.

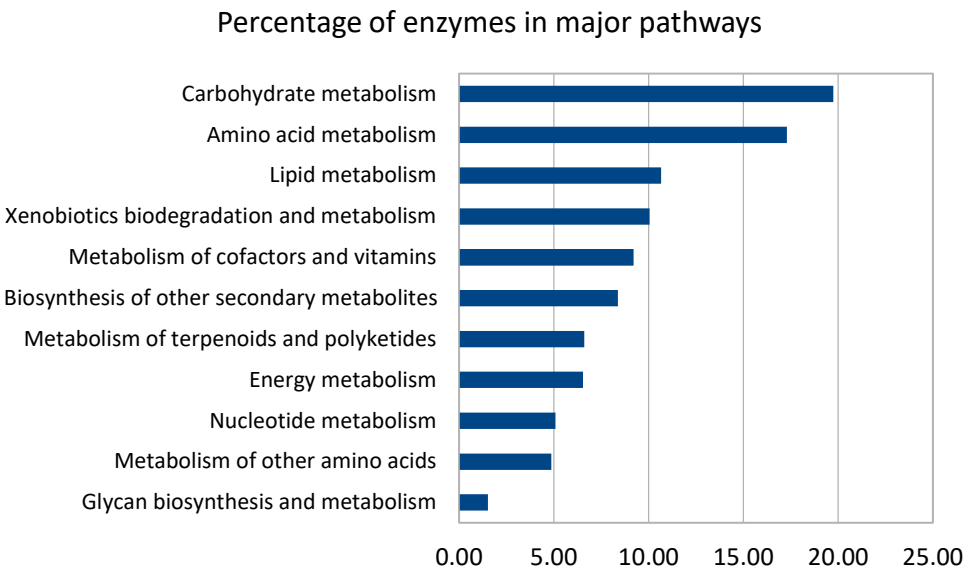

Figure 6.1 – Percentage of enzymes in major pathways of A. oryzae

The resulted network of *A.oryzae* organises in 11 major metabolic functions, as shown in Figure 6.1. The biggest amount of enzymes goes into carbohydrate metabolic processes, with 19.8% of enzymes. The fungus has many enzymes involving in lipid metabolism. This metabolic functional category has 10.7% of total enzymes and is ranked 3rd in the list.

6.3.3 Filling gaps by MeGaFiller

Table 6.3 shows statistics of gaps that are putatively filled for *A. oryzae*.

Table 6.3 – Filling gaps using MeGaFiller

Number of unique metabolic gaps	*376*
Number of metabolic gaps putatively filled	*93*
Number of putative candidates	*202*
Percentage of metabolic gaps filled	*24.7*

As can be seen, MeGaFiller can filled 93 gaps that NetA and EnzDP missed. These numbers are accounted for 24.7% of the current metabolic gaps. The number of candidates proposed by MeGaFiller for the gaps is 202, which is 2.2 candidates per gap, on average. Interestingly, if the threshold for EnzDP is set lower at 0.20, 11% of the current gaps can be recovered. Figure 6.2 in the next section shows an example of gap filling result.

6.3.4 Example of finding minimal pathway

This section illustrates an example of finding minimal pathway by MiniPath algorithm. MiniPath was run on Folate biosynthesis pathway using annotation made by EnzDP for *A. oryzae*. The resulted minimal pathway is shown on Figure 6.2. The pathway is connected, i.e. any possible output metabolite is reachable from input metabolites. The reactions in white are those in template pathway, but were removed by NetA because they neither have enzyme annotation nor connectivity dependency. Those reactions can safely be deleted from the pathway without making the pathway disconnected.

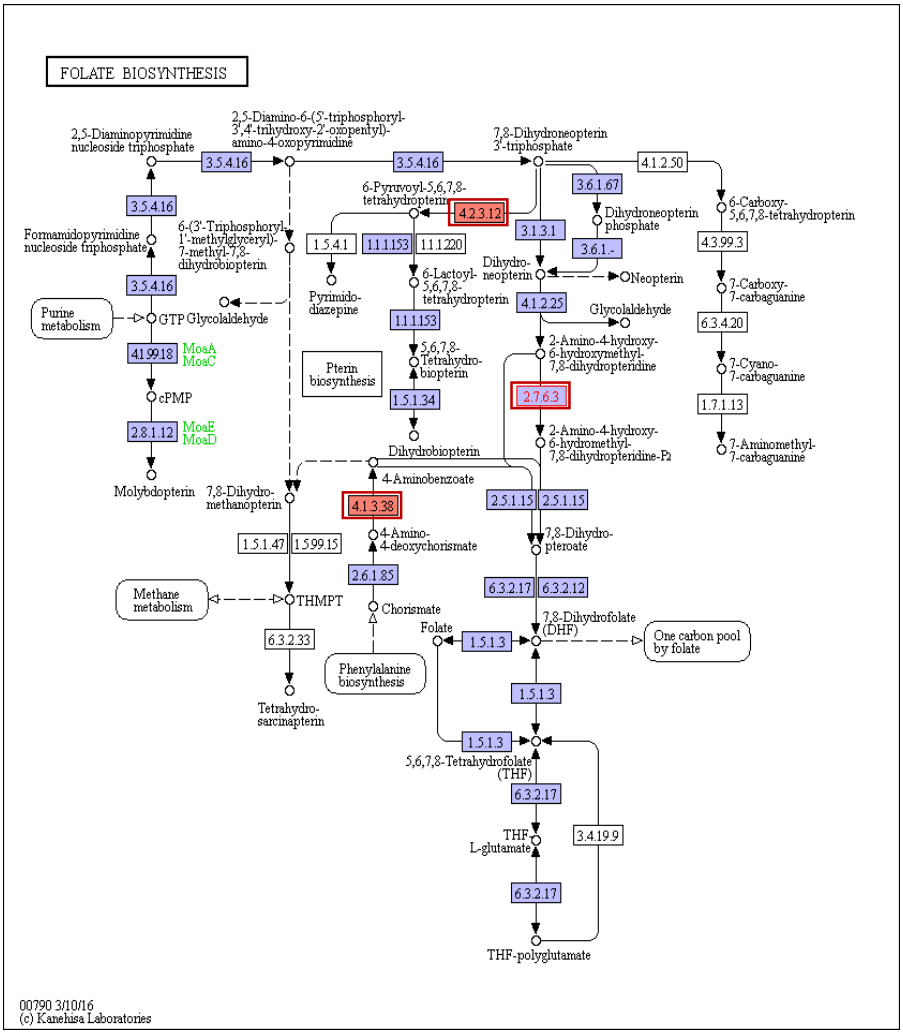

Figure 6.2 – Example of pathway predicted by NetA for *A. oryzae*

This 00790 pathway was predicted with 26 reactions, including 3 gaps (orange and pink) and 23 reactions (blue) covered by enzyme genes. One of the gap (2.7.6.3 – pink) was later filled by MeGaFiller-2.White reaction are those in template pathway but were removed by NetA due to no support evidence.

The reactions in blue are those with enzyme evidence. The two red reactions (4.2.3.12 and 4.1.3.38) are metabolic gaps as currently they have no enzyme gene support, but they are critical to the network connectivity. The pink reaction (2.7.6.3) was initially a metabolic gap, but latter was filled by MeGaFiller. Manual inspection shows that, EnzDP missed enzyme for 2.7.6.3, but this reaction is required for network connectivity. MeGaFiller predicted the enzyme for this reaction as AO090011000631. In fact, EnzDP assigned the enzyme AO090011000631 to the 4.1.2.25 activity, but MeGaFiller predicted that the enzyme has both functions. Other annotation sources also confirmed this MeGaFiller prediction. If the likelihood cut-off for EnzDP is set lower (at 0.20, lower than the default value of 0.4), this assignment will show up.

In this example, result of MiniPath is optimal. But in general, MiniPath is an approximation algorithm, with an approximation ratio of h_p, where $1/h_p$ is the pathway presence ratio. The example also shows that MeGaFiller is useful in filling gap when EnzDP misses a prediction due to cut-off filtering.

6.3.5 Comparison to *iWV1314*

Compared to iWV1314 network, the network built by NetA contains more reactions and enzymes. However, there is a significant discrepancy between the two networks. Figure 6.3 shows a comparison in three categories. While they agreed mostly on the unique reaction set, the enzyme set and gene-reaction pair set showed relatively lower agreement.

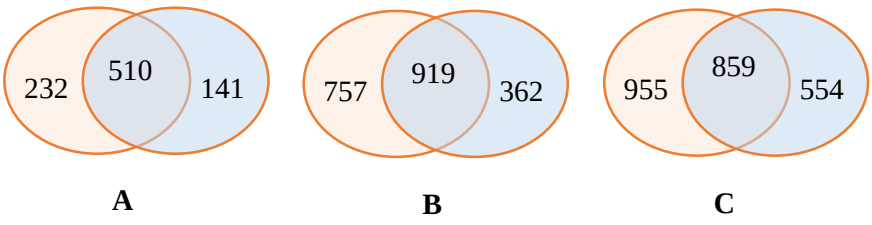

Figure 6.3 – Compare result of NetA on *A. oryzae* (right) to *iWV1314* network (left)
Note: *A*: Compare unique reaction sets. *B*: Compare enzyme sets. *C*: Compare gene-reaction pair sets.

The discrepancy between NetA's result and *iWV1314* network can be explained that their gene annotations have small overlap. The Jaccard score between the two initial enzyme sets is

only 0.4, which is low. This indicates that the discrepancy between them is significant. Or in other words, metabolic knowledge of *A.oryzae* is still significantly missing.

6.4 Discussion and Conclusions

In this application, the whole reconstruction pipeline was used to reconstruct a metabolic network of *A. oryzae*. The resulting network contains 742 unique reactions, across over 119 pathways in 11 major metabolic functions. On average over all predicted pathways, 72.4% of reactions have enzyme genes annotated. In comparison with *iWV1314* network, the result of NetA on *A. oryzae* mostly agrees on unique reaction set. However, there is a significant discrepancy in enzyme set, and in gene-reaction pair set.

The results also show that, there is a significant number (27.6%) of reactions that have no gene annotation. Those reactions are required for pathways connectivity, but currently no homologous genes are identified in *A. oryzae*. This is due to two main reasons: first, the failure of MiniPath algorithm and second, the incompleteness of annotations. Currently MiniPath does not guarantee optimal solutions. Our manual inspection of the result of MiniPath showed that, there many cases it could only produce local optimal. Meanwhile, the second reason is independent to the method, because there is no annotation data available to fill metabolic gaps in the resulting network. Our analysis on the initial enzyme annotation set of *A. oryzae* showed that, there is a significant inconsistency among current resources of *A. oryzae* genome annotations, including KEGG, Uniprot, DOGAN, AsperCyc databases. The computational predictions made by EnzDP and MeGaFiller also differ from those resources. This suggests that current metabolic knowledge of *A. oryzae* is far from complete. Thanks to re-assembling the metabolic network, this previously unknown fact is signified.

Chapter 7

Conclusions and future works

7.1 Conclusions

Metabolic network is a powerful systems biology tool for modelling and investigating cellular metabolism. With the availability of complete genome sequencing, genome annotation, pathways reference databases, as well as biochemical knowledge that is accumulating in public databases, reconstructing a whole genome scale metabolic network becomes feasible. In the last decade, metabolic networks of hundreds species have been reconstructed. Many computational methods have been developed to aid network reconstruction. However, there still has a significant number of gaps in these reconstructed networks. In addition, metabolic networks reconstructed by different sources have low agreement in their content. These issues indicate that those reconstructed networks are of low quality and not complete.

To build a high quality genome scale metabolic networks, a comprehensive protocol has recently been proposed. The reconstruction process involves several stages, including enzyme annotation, network assembly, gap filling, and network modeling and validation. Notably, reconstruction process is time consuming and may take very long time to be done. Significantly, there is a lot of manual curation involved. These currently are bottleneck issues in network reconstruction.

From the general reconstruction protocol, we have addressed three problems to improve network quality as well as to reduce time and effort requiring in building such network. The first problem is to reliably annotate the set of enzyme encoding genes in a given genome. This

problem is well established and widely known. However, while previous solutions focus on achieving a good precision, the predictable enzyme coverage needs further improvement. The second problem is to fill gaps after the network has initially been reconstructed. This problem involves predicting enzyme genes for existing metabolic reactions in the networks. The third problem is to quickly assembly a draft network, making use of reliable annotation resources. This procedure is important for updating new network as well as for rebuilding network from scratch. By addressing these three problems and developing computational methods for resolving them, network reconstruction become more productive and the human effort involved can be reduced.

For the gap filling problem, we have developed MeGaFiller to fill metabolic gaps in reconstructed networks. MeGaFiller, which based on retrofitting and integrating multiple function predictors into gap filler, can indeed overcome the issues of poorly characterized enzyme families faced by previous homology based methods. Our method was able to predict candidates for "difficult-to-fill" gaps in reconstructed networks which previous methods failed. As a result, about 35% of current gaps in several networks has been proposed at least one candidate.

For the enzyme annotation problem, we have developed EnzDP – an automated reliable enzyme function predictor. This is a novel strategy for enzyme classification based on the weighted architecture of functional domains (DEAS) and calibrated HMM profiles. Overall our bottom-up method had significantly improved accuracy and coverage over direct top-down methods. EnzDP achieved 94.5% micro-accuracy in 5-fold cross validation and outperformed other alternatives. EnzDP can serve as fast reliable enzyme classifier for timely analyze genomics and metagenomics sequence data.

For network assembly problem, we developed NetA to assembly a connected network without reaction gaps, with minimized number of metabolic gaps. Input of NetA is a set of annotated enzymes, which can be produced by EnzDP. We showed that the network assembly problem is NP-Hard, and developed NetA as an approximation algorithm. Following graph-theoretic approach, NetA predicts and optimizes template pathways by deleting unnecessary

reactions while maintaining connectivity. We applied EnzDP, NetA and MeGaFiller as a combined tool on rebuilding *A.oryzae* network, which resulted in an improved enzyme coverage, and reliable predictions. Recently, this initial pipeline was partially applied on *Cordyceps militaris*, which resulted in the first genome-scale metabolic network representing cellular metabolisms of *C. militaris* (iWV1170) [116].

7.2 Summary of contributions

In this research, a proof of concept methodology to improve the genome-scale metabolic network reconstruction pipeline has been developed. The combination of developed methods (EnzDP, NetA, and MeGaFiller) serves as a quick and effective tool to reduce time and human involvement in reconstructing high quality networks. This will push further the research on genome-scale reconstructions, which potentially leads to plentiful applications in medicine, bioengineering, and bio-energy/food processing industry.

Three contributions are:

- ✓ A novel method for finding missing gene and gap filling problem, which can improve quality of reconstructed networks, as well as help enriching metabolic annotation;
- ✓ A stringent high throughput reliable enzyme classifier for annotating genomics sequences. This high throughput method is essential to timely interpret the huge amount of genomics data. The predictor produces high quality annotation data, which initiates other metabolic/genomics investigations. As a direct result, the method helps to reduce reconstruction errors towards a high quality metabolic reconstruction;
- ✓ A quick and effective network assembly method with no reaction gaps and minimum number of metabolic gaps, to reduce time and manual effort in genome-scale metabolic reconstruction. Making use of gap filling, enzyme function prediction, and network assembly methods, this combined reconstruction aid tool will lead to new or updated metabolic networks as direct applications.

7.3 Future works

Though we have made every effort to improve the network reconstruction process which resulted in developed stringent tools, there is still room for further improvement towards a comprehensive and complete genome-scale reconstruction. This section points out some of directions that can be considered for future works.

7.3.1 Pathway predictions and optimizations

On doing network assembly, we find that available resources is not consistent and not fully complete, which significantly affect the reconstruction quality. Thus better algorithms for network assembly with less dependence on availability of annotation data are essential.

An improvement to consider is to break down template pathways into smaller viable branches. Currently, we use template pathways that contain all possible metabolic transformations that can happen in the pathway, across all known organisms. However, an organism-specific pathway may express only a small part of it. Our analysis on KEGG database showed that, only about 20% of reactions in templates are in fact occurred in *S. cerevisiae* and *E. coli*. Therefore, general template pathways may contain transformations that are irrelevant to the genome of interest. This issue complicates the MiniPath algorithm, since the algorithm has to connect those irrelevant pathway branches. So, to improve overall result, better set of template pathways should be use. Beside KEGG pathways, BioCyc database is a good alternative.

On the other hand, MiniPath is currently an approximation algorithm, which has faced many sub-optimal cases. Thus, there is still a lot of room for improving MiniPath with optimal algorithm, e.g., with good heuristics.

7.3.2 Network refinement and model simulation

NetA currently produces a back-bone network of reactions, which is referred to as a draft reconstruction. To refine the network, the tasks in the pipeline should perform iteratively, with help of the combined tool. Indeed, to transform the network into usable model, more information should be added into the network. Essentially, network reconstruction should

proceed to the phase of the pipeline to verify those predicted metabolic pathways. Information such as transcriptome/expression data is essential to improve network reliability

7.3.3 Transport proteins

Transporters are proteins that serve the function of transporting chemicals within cellular environment across biological membranes. They are important to metabolic processes as they move ions, micro and macro molecules, thus connecting metabolic pathways between cell compartments. With expansion of transporters, a metabolic pathway/network becomes much more comprehensive and integrative, which potentially leads to more useful systems biology applications. Therefore, it is essential to annotate transporter functions expressed in the genome, and add them into the network. Predicting transporter functions from genomics data is also a challenging problem to be explored.

Bibliography

1. Förster, J., et al., *Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network.* Genome Res, 2003. **13**(2): p. 244-53.
2. Nookaew, I., et al., *The genome-scale metabolic model iIN800 of Saccharomyces cerevisiae and its validation: a scaffold to query lipid metabolism.* BMC Syst Biol, 2008. **2**: p. 71.
3. Vongsangnak, W., et al., *Improved annotation through genome-scale metabolic modeling of Aspergillus oryzae.* BMC Genomics, 2008. **9**: p. 245.
4. Vongsangnak, W., et al., *Genome-scale analysis of the metabolic networks of oleaginous Zygomycete fungi.* Gene, 2013. **521**(1): p. 180-90.
5. David, H., et al., *Analysis of Aspergillus nidulans metabolism at the genome-scale.* BMC Genomics, 2008. **9**: p. 163.
6. Andersen, M.R., M.L. Nielsen, and J. Nielsen, *Metabolic model integration of the bibliome, genome, metabolome and reactome of Aspergillus niger.* Mol Syst Biol, 2008. **4**: p. 178.
7. DeJongh, M., et al., *Toward the automated generation of genome-scale metabolic networks in the SEED.* BMC Bioinformatics, 2007. **8**: p. 139.
8. Notebaart, R.A., et al., *Accelerating the reconstruction of genome-scale metabolic networks.* BMC Bioinformatics, 2006. **7**: p. 296.
9. Herrgård, M.J., et al., *A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology.* Nat Biotechnol, 2008. **26**(10): p. 1155-60.
10. Swainston, N., P. Mendes, and D.B. Kell, *An analysis of a 'community-driven' reconstruction of the human metabolic network.* Metabolomics, 2013. **9**(4): p. 757-764.
11. Hanson, A.D., et al., *'Unknown' proteins and 'orphan' enzymes: the missing half of the engineering parts list--and how to find it.* Biochem J, 2010. **425**(1): p. 1-11.
12. Osterman, A. and R. Overbeek, *Missing genes in metabolic pathways: a comparative genomics approach.* Curr Opin Chem Biol, 2003. **7**(2): p. 238-51.
13. Thiele, I. and B. Palsson, *A protocol for generating a high-quality genome-scale metabolic reconstruction.* Nat Protoc, 2010. **5**(1): p. 93-121.
14. Green, M.L. and P.D. Karp, *Genome annotation errors in pathway databases due to semantic ambiguity in partial EC numbers.* Nucleic Acids Res, 2005. **33**(13): p. 4035-9.
15. Poptsova, M.S. and J.P. Gogarten, *Using comparative genome analysis to identify problems in annotated microbial genomes.* Microbiology, 2010. **156**(Pt 7): p. 1909-17.
16. Jones, C.E., A.L. Brown, and U. Baumann, *Estimating the annotation error rate of curated GO database sequence annotations.* BMC Bioinformatics, 2007. **8**: p. 170.
17. Durot, M., P.Y. Bourguignon, and V. Schachter, *Genome-scale models of bacterial metabolism: reconstruction and applications.* FEMS Microbiol Rev, 2009. **33**(1): p. 164-90.
18. Barrett, A.J., *Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). Enzyme Nomenclature. Recommendations 1992. Supplement 4: corrections and additions (1997).* Eur J Biochem, 1997. **250**(1): p. 1-6.
19. Koutinas, M., et al., *Bioprocess systems engineering: transferring traditional process engineering principles to industrial biotechnology.* Comput Struct Biotechnol J, 2012. **3**: p. e201210022.
20. Abe, K., et al., *Impact of Aspergillus oryzae genomics on industrial production of metabolites.* Mycopathologia, 2006. **162**(3): p. 143-53.

21. Burgal, J., et al., *Metabolic engineering of hydroxy fatty acid production in plants: RcDGAT2 drives dramatic increases in ricinoleate levels in seed oil.* Plant Biotechnol J, 2008. **6**(8): p. 819-31.
22. Minty, J.J., et al., *Design and characterization of synthetic fungal-bacterial consortia for direct production of isobutanol from cellulosic biomass.* Proc Natl Acad Sci U S A, 2013. **110**(36): p. 14592-7.
23. Nikoloski, Z., et al., *Metabolic networks are NP-hard to reconstruct.* J Theor Biol, 2008. **254**(4): p. 807-16.
24. Magrane, M. and U. Consortium, *UniProt Knowledgebase: a hub of integrated protein data.* Database (Oxford), 2011. **2011**: p. bar009.
25. Consortium, U., *Reorganizing the protein space at the Universal Protein Resource (UniProt).* Nucleic Acids Res, 2012. **40**(Database issue): p. D71-5.
26. Punta, M., et al., *The Pfam protein families database.* Nucleic Acids Res, 2012. **40**(Database issue): p. D290-301.
27. Finn, R.D., et al., *Pfam: the protein families database.* Nucleic Acids Res, 2014. **42**(Database issue): p. D222-30.
28. Marchler-Bauer, A., et al., *CDD: a database of conserved domain alignments with links to domain three-dimensional structure.* Nucleic Acids Res, 2002. **30**(1): p. 281-3.
29. Marchler-Bauer, A., et al., *CDD: conserved domains and protein three-dimensional structure.* Nucleic Acids Res, 2013. **41**(Database issue): p. D348-52.
30. Kanehisa, M., *The KEGG database.* Novartis Found Symp, 2002. **247**: p. 91-101; discussion 101-3, 119-28, 244-52.
31. Kanehisa, M., et al., *KEGG for integration and interpretation of large-scale molecular data sets.* Nucleic Acids Res, 2012. **40**(Database issue): p. D109-14.
32. Tanabe, M. and M. Kanehisa, *Using the KEGG database resource.* Curr Protoc Bioinformatics, 2012. **Chapter 1**: p. Unit1.12.
33. Dimmer, E.C., et al., *The UniProt-GO Annotation database in 2011.* Nucleic Acids Res, 2012. **40**(Database issue): p. D565-70.
34. Hunter, S., et al., *InterPro: the integrative protein signature database.* Nucleic Acids Res, 2009. **37**(Database issue): p. D211-5.
35. Laskowski, R.A., V.V. Chistyakov, and J.M. Thornton, *PDBsum more: new summaries and analyses of the known 3D structures of proteins and nucleic acids.* Nucleic Acids Res, 2005. **33**(Database issue): p. D266-8.
36. Claudel-Renard, C., et al., *Enzyme-specific profiles for genome annotation: PRIAM.* Nucleic Acids Res, 2003. **31**(22): p. 6633-9.
37. Tian, W., A.K. Arakaki, and J. Skolnick, *EFICAz: a comprehensive approach for accurate genome-scale enzyme function inference.* Nucleic Acids Res, 2004. **32**(21): p. 6226-39.
38. Arakaki, A.K., Y. Huang, and J. Skolnick, *EFICAz2: enzyme function inference by a combined approach enhanced by machine learning.* BMC Bioinformatics, 2009. **10**: p. 107.
39. Desai, D.K., et al., *ModEnzA: Accurate Identification of Metabolic Enzymes Using Function Specific Profile HMMs with Optimised Discrimination Threshold and Modified Emission Probabilities.* Adv Bioinformatics, 2011. **2011**: p. 743782.
40. Schomburg, I., et al., *BRENDA in 2013: integrated reactions, kinetic data, enzyme function data, improved disease classification: new options and contents in BRENDA.* Nucleic Acids Res, 2013. **41**(Database issue): p. D764-72.
41. Caspi, R., et al., *The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases.* Nucleic Acids Res, 2012. **40**(Database issue): p. D742-53.
42. Machida, M., et al., *Genome sequencing and analysis of Aspergillus oryzae.* Nature, 2005. **438**(7071): p. 1157-61.
43. Friedberg, I., *Automated protein function prediction--the genomic challenge.* Brief Bioinform, 2006. **7**(3): p. 225-42.

44. Rentzsch, R. and C.A. Orengo, *Protein function prediction--the power of multiplicity.* Trends Biotechnol, 2009. **27**(4): p. 210-9.
45. Mohammed, A. and C. Guda, *Computational Approaches for Automated Classification of Enzyme Sequences.* J Proteomics Bioinform, 2011. **4**: p. 147-152.
46. Rost, B., *Enzyme function less conserved than anticipated.* J Mol Biol, 2002. **318**(2): p. 595-608.
47. Tian, W. and J. Skolnick, *How well is enzyme function conserved as a function of pairwise sequence identity?* J Mol Biol, 2003. **333**(4): p. 863-82.
48. Liu, B., et al., *Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection.* Bioinformatics, 2014. **30**(4): p. 472-9.
49. Shen, H.B. and K.C. Chou, *EzyPred: a top-down approach for predicting enzyme functional classes and subclasses.* Biochem Biophys Res Commun, 2007. **364**(1): p. 53-9.
50. Forslund, K. and E.L. Sonnhammer, *Predicting protein function from domain content.* Bioinformatics, 2008. **24**(15): p. 1681-7.
51. Messih, M.A., et al., *Protein domain recurrence and order can enhance prediction of protein functions.* Bioinformatics, 2012. **28**(18): p. i444-i450.
52. Sigrist, C.J., et al., *New and continuing developments at PROSITE.* Nucleic Acids Res, 2013. **41**(Database issue): p. D344-7.
53. Yu, C., et al., *Genome-wide enzyme annotation with precision control: catalytic families (CatFam) databases.* Proteins, 2009. **74**(2): p. 449-60.
54. Nguyen, N.N., et al., *EnzDP: improved enzyme annotation for metabolic network reconstruction based on domain composition profiles.* J Bioinform Comput Biol, 2015. **13**(5): p. 1543003.
55. Zhu, J. and Z. Weng, *FAST: a novel protein structure alignment algorithm.* Proteins, 2005. **58**(3): p. 618-27.
56. Ye, Y. and A. Godzik, *Flexible structure alignment by chaining aligned fragment pairs allowing twists.* Bioinformatics, 2003. **19 Suppl 2**: p. ii246-55.
57. Madej, T., J.F. Gibrat, and S.H. Bryant, *Threading a database of protein cores.* Proteins, 1995. **23**(3): p. 356-69.
58. Novotny, M., D. Madsen, and G.J. Kleywegt, *Evaluation of protein fold comparison servers.* Proteins, 2004. **54**(2): p. 260-70.
59. Porter, C.T., G.J. Bartlett, and J.M. Thornton, *The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data.* Nucleic Acids Res, 2004. **32**(Database issue): p. D129-33.
60. Furnham, N., et al., *The Catalytic Site Atlas 2.0: cataloging catalytic sites and residues identified in enzymes.* Nucleic Acids Res, 2014. **42**(Database issue): p. D485-9.
61. George, R.A., et al., *Effective function annotation through catalytic residue conservation.* Proc Natl Acad Sci U S A, 2005. **102**(35): p. 12299-304.
62. Sharan, R., I. Ulitsky, and R. Shamir, *Network-based prediction of protein function.* Mol Syst Biol, 2007. **3**: p. 88.
63. Janga, S.C., J.J. Díaz-Mejía, and G. Moreno-Hagelsieb, *Network-based function prediction and interactomics: the case for metabolic enzymes.* Metab Eng, 2011. **13**(1): p. 1-10.
64. Espadaler, J., et al., *Prediction of enzyme function by combining sequence similarity and protein interactions.* BMC Bioinformatics, 2008. **9**: p. 249.
65. Pellegrini, M., et al., *Assigning protein functions by comparative genome analysis: protein phylogenetic profiles.* Proc Natl Acad Sci U S A, 1999. **96**(8): p. 4285-8.
66. Overbeek, R., et al., *The use of gene clusters to infer functional coupling.* Proc Natl Acad Sci U S A, 1999. **96**(6): p. 2896-901.
67. Moreno-Hagelsieb, G. and S.C. Janga, *Operons and the effect of genome redundancy in deciphering functional relationships using phylogenetic profiles.* Proteins, 2008. **70**(2): p. 344-52.

68. Enright, A.J., et al., *Protein interaction maps for complete genomes based on gene fusion events.* Nature, 1999. **402**(6757): p. 86-90.
69. Tarassov, K., et al., *An in vivo map of the yeast protein interactome.* Science, 2008. **320**(5882): p. 1465-70.
70. Luo, F., et al., *Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory.* BMC Bioinformatics, 2007. **8**: p. 299.
71. Ruan, J., A.K. Dean, and W. Zhang, *A general co-expression network-based approach to gene expression analysis: comparison and applications.* BMC Syst Biol, 2010. **4**: p. 8.
72. Chatr-Aryamontri, A., et al., *The BioGRID interaction database: 2013 update.* Nucleic Acids Res, 2013. **41**(Database issue): p. D816-23.
73. Pagel, P., et al., *The MIPS mammalian protein-protein interaction database.* Bioinformatics, 2005. **21**(6): p. 832-4.
74. Franceschini, A., et al., *STRING v9.1: protein-protein interaction networks, with increased coverage and integration.* Nucleic Acids Res, 2013. **41**(Database issue): p. D808-15.
75. Bowers, P.M., et al., *Prolinks: a database of protein functional linkages derived from coevolution.* Genome Biol, 2004. **5**(5): p. R35.
76. Warde-Farley, D., et al., *The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function.* Nucleic Acids Res, 2010. **38**(Web Server issue): p. W214-20.
77. Chua, H.N., W.K. Sung, and L. Wong, *An efficient strategy for extensive integration of diverse biological data for protein function prediction.* Bioinformatics, 2007. **23**(24): p. 3364-73.
78. Hishigaki, H., et al., *Assessment of prediction accuracy of protein function from protein--protein interaction data.* Yeast, 2001. **18**(6): p. 523-31.
79. Kotera, M., et al., *GENIES: gene network inference engine based on supervised analysis.* Nucleic Acids Res, 2012. **40**(Web Server issue): p. W162-7.
80. Erdin, S., A.M. Lisewski, and O. Lichtarge, *Protein function prediction: towards integration of similarity metrics.* Curr Opin Struct Biol, 2011. **21**(2): p. 180-8.
81. Karp, P.D., S. Paley, and P. Romero, *The Pathway Tools software.* Bioinformatics, 2002. **18 Suppl 1**: p. S225-32.
82. Green, M.L. and P.D. Karp, *A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases.* BMC Bioinformatics, 2004. **5**: p. 76.
83. Pitkänen, E., P. Jouhten, and J. Rousu, *Inferring branching pathways in genome-scale metabolic networks.* BMC Syst Biol, 2009. **3**: p. 103.
84. Orth, J.D. and B. Palsson, *Gap-filling analysis of the iJO1366 Escherichia coli metabolic network reconstruction for discovery of metabolic functions.* BMC Syst Biol, 2012. **6**: p. 30.
85. Satish Kumar, V., M.S. Dasika, and C.D. Maranas, *Optimization based automated curation of metabolic reconstructions.* BMC Bioinformatics, 2007. **8**: p. 212.
86. Kumar, V.S. and C.D. Maranas, *GrowMatch: an automated method for reconciling in silico/in vivo growth predictions.* PLoS Comput Biol, 2009. **5**(3): p. e1000308.
87. Pitkänen, E., et al., *Comparative genome-scale reconstruction of gapless metabolic networks for present and ancestral species.* PLoS Comput Biol, 2014. **10**(2): p. e1003465.
88. Reed, J.L., et al., *An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR).* Genome Biol, 2003. **4**(9): p. R54.
89. Yamanishi, Y., et al., *Prediction of missing enzyme genes in a bacterial metabolic network. Reconstruction of the lysine-degradation pathway of Pseudomonas aeruginosa.* FEBS J, 2007. **274**(9): p. 2262-73.
90. Kharchenko, P., D. Vitkup, and G.M. Church, *Filling gaps in a metabolic network using expression information.* Bioinformatics, 2004. **20 Suppl 1**: p. i178-85.
91. Chen, L. and D. Vitkup, *Predicting genes for orphan metabolic activities using phylogenetic profiles.* Genome Biol, 2006. **7**(2): p. R17.

92. Kharchenko, P., et al., *Identifying metabolic enzymes with multiple types of association evidence.* BMC Bioinformatics, 2006. **7**: p. 177.
93. Yamada, T., et al., *Prediction and identification of sequences coding for orphan enzymes using genomic and metagenomic neighbours.* Mol Syst Biol, 2012. **8**: p. 581.
94. Borodina, I., P. Krabben, and J. Nielsen, *Genome-scale analysis of Streptomyces coelicolor A3(2) metabolism.* Genome Res, 2005. **15**(6): p. 820-9.
95. Martin, D.M., M. Berriman, and G.J. Barton, *GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes.* BMC Bioinformatics, 2004. **5**: p. 178.
96. Hawkins, T., et al., *PFP: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data.* Proteins, 2009. **74**(3): p. 566-82.
97. Hawkins, T., S. Luban, and D. Kihara, *Enhanced automated function prediction using distantly related sequences and contextual association by PFP.* Protein Sci, 2006. **15**(6): p. 1550-6.
98. Conesa, A., et al., *Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research.* Bioinformatics, 2005. **21**(18): p. 3674-6.
99. Conesa, A. and S. Götz, *Blast2GO: A comprehensive suite for functional analysis in plant genomics.* Int J Plant Genomics, 2008. **2008**: p. 619832.
100. Cvijovic, M., et al., *BioMet Toolbox: genome-wide analysis of metabolism.* Nucleic Acids Res, 2010. **38**(Web Server issue): p. W144-9.
101. Cherry, J.M., et al., *Saccharomyces Genome Database: the genomics resource of budding yeast.* Nucleic Acids Res, 2012. **40**(Database issue): p. D700-5.
102. Cerqueira, G.C., et al., *The Aspergillus Genome Database: multispecies curation and incorporation of RNA-Seq data to improve structural gene annotations.* Nucleic Acids Res, 2014. **42**(Database issue): p. D705-10.
103. Bentley, S.D., et al., *Complete genome sequence of the model actinomycete Streptomyces coelicolor A3(2).* Nature, 2002. **417**(6885): p. 141-7.
104. Nguyen, N.N., et al., *Megafiller: A Retrofitted Protein Function Predictor for Filling Gaps in Metabolic Networks.* J Proteomics Bioinform S9-003, 2015. **S9**(003).
105. Gerlt, J.A., et al., *The Enzyme Function Initiative.* Biochemistry, 2011. **50**(46): p. 9950-62.
106. Cai, Y.D. and K.C. Chou, *Using functional domain composition to predict enzyme family classes.* J Proteome Res, 2005. **4**(1): p. 109-11.
107. Geer, L.Y., et al., *CDART: protein homology by domain architecture.* Genome Res, 2002. **12**(10): p. 1619-23.
108. Chen, T.W., et al., *DODO: an efficient orthologous genes assignment tool based on domain architectures. Domain based ortholog detection.* BMC Bioinformatics, 2010. **11 Suppl 7**: p. S6.
109. Furnham, N., et al., *Missing in action: enzyme functional annotations in biological databases.* Nat Chem Biol, 2009. **5**(8): p. 521-5.
110. Eddy, S.R., *Accelerated Profile HMM Searches.* PLoS Comput Biol, 2011. **7**(10): p. e1002195.
111. Sievers, F., et al., *Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega.* Mol Syst Biol, 2011. **7**: p. 539.
112. Liu, B., et al., *Identification of real microRNA precursors with a pseudo structure status composition approach.* PLoS One, 2015. **10**(3): p. e0121501.
113. Liu, B., et al., *iMiRNA-PseDPC: microRNA precursor identification with a pseudo distance-pair composition approach.* J Biomol Struct Dyn, 2015: p. 1-13.
114. Liu, B., et al., *miRNA-dis: microRNA precursor identification based on distance structure status pairs.* Mol Biosyst, 2015. **11**(4): p. 1194-204.
115. Zhao, G., et al., *Comparative genomic analysis of Aspergillus oryzae strains 3.042 and RIB40 for soy sauce fermentation.* Int J Food Microbiol, 2013. **164**(2-3): p. 148-54.
116. Vongsangnak, W., et al., *Genome-scale metabolic network of Cordyceps militaris useful for comparative analysis of entomopathogenic fungi.* Gene, 2017.