Comparative and Evolutionary Analysis of Cellular Pathways

Manikandan Narayanan

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2007-141
http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-141.html

December 5, 2007

Comparative and Evolutionary Analysis of Cellular Pathways

by

Manikandan Narayanan

B.E. (Anna University, College of Engineering, Guindy, Chennai, India) 2001
M.S. (University of California, Berkeley) 2003

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science
with
Designated Emphasis in Computational and Genomic Biology

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Richard M. Karp, Chair
Professor Satish Rao
Professor Kimmen Sjölander

Fall 2007

The dissertation of Manikandan Narayanan is approved.

Chair Date

Date

Date

University of California, Berkeley

Fall 2007

Comparative and Evolutionary Analysis of Cellular Pathways

by

Manikandan Narayanan

Abstract

Comparative and Evolutionary Analysis of Cellular Pathways

by

Manikandan Narayanan

Doctor of Philosophy in Computer Science

with

Designated Emphasis in Computational and Genomic Biology

University of California, Berkeley

Professor Richard M. Karp, Chair

Various pathways maintain the structure, function and health of a cell, and intricate molecular mechanisms underlie these cellular pathways. Inquiring how such mechanisms could have evolved is a basic question in evolutionary biology with wide-ranging implications for predicting and altering cellular phenotypes. This dissertation presents our work on the computational analysis of genome-level data (on biomolecular sequences and interactions) available for many organisms to study the conservation and evolution of cellular pathways.

We study conservation of pathways in the context of comparative analysis of protein interaction networks. We specifically present a method based on a graph-matching algorithm to detect conserved pathways between two protein networks. Our algorithm is provably efficient unlike the search heuristics used in previous methods, and is novel in the broader field of graph-matching as well. We apply the method to compare the yeast network with the human, fruit fly and nematode worm networks, evaluate the detected conserved pathways using known yeast protein complexes, and demonstrate applications to function prediction.

We study evolution of pathways in the context of phylogenetic analysis of bacterial

and archaeal pathways. We specifically present a tractable probabilistic model for pathway evolution that makes the assumptions about pathway evolution explicit, in contrast to the few past studies that use discrete pathway similarity based models. We then apply the model to estimate the phylogeny along which a pathway such as citric acid cycle or chemotaxis evolved from its unknown ancestral forms to extant forms. We interpret the estimated phylogenies of such pathways involved in essential metabolisms or stress responses using known species phylogenies and published cellular phenotypes.

Professor Richard M. Karp
Dissertation Committee Chair

To amma, appa, akka and kudumbam.

Contents

List of Figures

List of Tables

Acknowledgements

The research studies presented in this thesis are joint work with co-workers at Berkeley. The first study on protein network comparison is joint work with Richard M. Karp, and the second on pathway phylogeny estimation is joint work with Amoolya H. Singh and Richard M. Karp. I would like to thank them for their contributions, feedback and patience. I have learnt much of computation and biology collaborating with them.

With regard to the first research project, we thank Jayanth Kumar Kannan for valuable discussions that led to the graph-matching algorithm. We thank Sourav Chatterji, Sridhar Rajagopalan and Ben Reichardt for comments at different stages of the project. We thank Roded Sharan, Silpa Suthram and Mehmet Koyutürk for help with their previous work. Our description of the project in this thesis overlaps somewhat with our published description (in the Journal of Computational Biology, volume 14(7), pages 892-907, year 2007).

With regard to the second research project, we thank David Bindel and Alistair Sinclair for their time discussing matrix exponentials, and Adam Arkin for feedback on the results. We thank Ian Holmes for numerous clarifying discussions about probabilistic models during meetings and course lectures. We thank Sonesh Surana for help with some data files that were crucial to getting the results. This project combines Amoolya Singh's major contribution to the results section (input data for microbial pathways, and visualization and biological interpretation of estimated phylogenies); design and implementation of the method, and its application to artificial pathways by myself, and Richard Karp's feedback on both results and methods aspects of the work.

My PhD work was partially supported by NSF Award 331494. I thank graduate staff La Shana Porlaris and Linda Stubbs for helping me with the logistics of the PhD program.

I have benefited greatly from interactions with stimulating teachers and professors throughout my education. My undergraduate mentor Ranjani Parthasarathi and project mates Ram Rangan and Satish Narayanasamy helped me consider and enter graduate school. A couple of theory courses, especially the first one I took under Satish Rao, healthy encouragement from my Masters advisor Katherine Yelick, a conversation with Sailesh

Krishnamurthy, and warm welcome by Richard Karp and the theory group students and faculty in general is all it took to convince me to pursue a PhD in applied theory after my Masters-only program. My post-graduate transition from Berkeley to Seattle has benefited much from interactions with Kimmen Sjölander, who emphasized the importance of orienting research towards current problems.

The single most influence on my research has of course been my advisor Richard Karp. I am deeply grateful to him for teaching and encouraging me patiently through the PhD program. His clear approach to problem-solving, experience in formulating fresh theoretical problems that are relevant to applied fields, methodical attention to the projects we worked on, and disciplined manner of teaching theory courses will continue to influence how I understand and do research.

I am a product of interactions with close members of my family and friends besides teachers. Any aspect of who I am or what I do is heavily influenced by the time they spend with me. Doing a PhD has involved combining all that they have provided me: strength, inspiration, encouragement, comfort, support, fun and distraction. I am truly and deeply thankful to every one of them. Instead of putting down their names, I hope the unspoken words here would more powerfully convey my gratitude to them.

Go Bears!

Chapter 1

Introduction

1.1 Subject of Inquiry

A diverse array of processes maintain cellular structure and function: metabolic reactions, DNA repair, protein translation, protein transport and signal transduction are just a few of them. Intricate and fascinating molecular mechanisms underlie cellular processes (Figure 1.1), and an inquiry into how such mechanisms could have evolved from some unknown ancestral forms and how much they are conserved across present-day organisms is very engaging. This inquiry addresses fundamental questions in evolutionary biology, and has broad implications for predicting or altering cellular behaviors.

This thesis presents the design and application of new computational methods to study some basic questions on the conservation and evolution of cellular processes. Our focus is on questions that are well-suited for study by computational analysis of genome-level data available for many organisms. Specifically, we present rigorous comparative and phylogenetic analysis of biomolecular sequence and interaction data pertaining to various cellular processes.

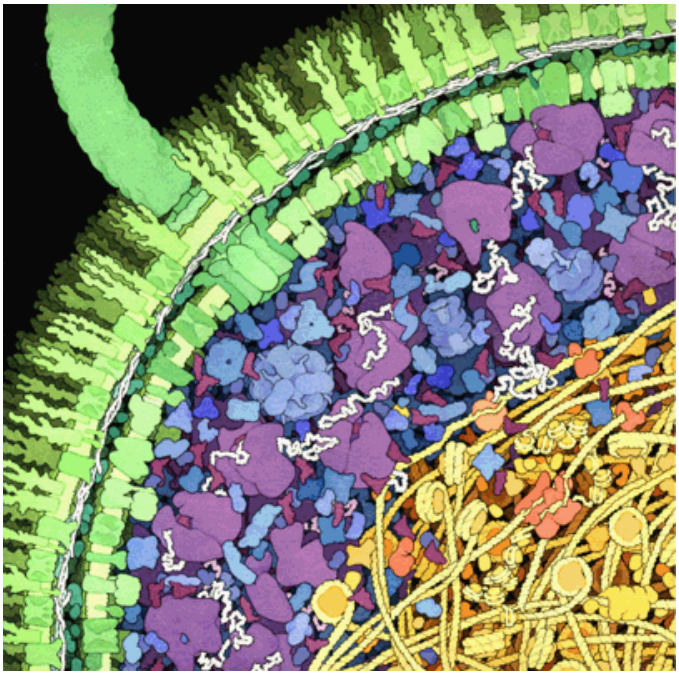

Figure 1.1. Illustration of a living cell by David Goodsell, researcher and artist at the Scripps Research Institute. The illustration (best viewed in color) magnifies a small cross-section of a bacterial *Escherichia coli* cell to show individual molecules involved in cellular processes. For example, ribosomes (large purple molecules) translating mRNA (white strands) into proteins with the help of tRNA (small, L-shaped maroon molecules) are shown. Another example shown in the center of the nucleoid (yellow and orange region) is a DNA polymerase (in red-orange) replicating new DNA at a replication fork. Besides polymerase, the process of replication involves several other proteins acting in a coordinated manner at the replication fork.

1.2 Genomic Data on Cellular Pathways

A computational inquiry into the evolutionary biology of cellular processes begins by abstracting the molecular mechanisms underlying these processes as pathways. A cellular pathway is a network of interacting genes, proteins and other biomolecules that are temporally coordinated to direct a specific cellular process. The interactions could be physical (DNA-protein, protein-protein, protein-small molecule) or genetic (functional or regulatory association between genes inferred from genetic experiments) with particular attributes such as transcriptional activation or repression, binding in a complex, phosphorylation or de-phosphorylation, methylation or de-methylation, etc. The biomolecules could have attributes such as gene sequences, expression levels, etc.

Series of detailed genetic and biochemical studies on specific pathways have produced a wealth of information on the molecular interactions and mechanisms that constitute the pathways (Alberts *et al.*, 2002). More recently, high-throughput though noisy experiments have greatly increased the number of observed cell-wide interactions. For instance, yeast two-hybrid or tandem access purification experiments (Bork *et al.*, 2004) have revealed roughly 15000 cell-wide interactions among about 4000 yeast proteins. The combination of these pathway data, i.e., reliable interaction data on well-studied pathways and large-scale data on cell-wide interactions, is a significant complement to biomolecular sequence data. The availability of these data enhances the study of interplay among network structure, function and evolution of cellular processes.

1.3 Evolution of Cellular Pathways

Pathways explain cellular phenotypes (observable behaviors) better than individual molecules, as most cellular processes result from the combined effort of more than one gene or protein. Hence a key motivation to study pathway evolution is to better explain the genetic basis of phenotypic variation, the ultimate challenge in evolutionary biology. Further, important applications that predict or alter a cellular phenotype (pathway function) can benefit from knowing the evolutionary diversity of the relevant pathway i.e., the variation in the components of this pathway across species and its correlation if any with phenotypic variation.

1.3.1 Specific Case Studies

We sample current knowledge on the evolution of two specific pathways, based on information in two nice reviews. The intent is to provide a context for works that trace the evolution of a pathway using concerted changes in the components (molecules and interactions) of the pathway over evolutionary time.

Protein transport into mitochondria (Dolezal *et al.*, 2006). The origin of organelles such as mitochondria is an important event in the evolution of eukaryotic cells. According to the endosymbiotic theory, symbiotic bacteria transformed to mitochondria in ancestral eukaryotes. A transport pathway also evolved to direct nuclear-encoded proteins into mitochondria. Some modules in this transport pathway such as molecular chaperones and signal peptidases appear to have evolved from the symbiotic bacteria's transport mechanisms, whereas certain other modules in this pathway such as the translocase complexes appear to have originated *de novo* in the ancestor of all eukaryotes. These hypotheses are based on the cross-species conservation of the pathway genes detected via sensitive sequence searches of all sequenced eukaryotic genomes, and in some cases on the phylogeny of the homologs of a gene (Dolezal *et al.*, 2006).

Innate immune response pathway (Kimbrell and Beutler, 2001). Innate immune response refers to mechanisms that defend multicellular organisms against microbial infections. The Toll pathway is a prominent example that responds to certain fungal and bacterial infections using a family of Toll receptors, downstream signaling proteins such as Tube/Pelle in fruit fly, and target antimicrobial proteins. These components are conserved between fruit fly and mice, but some receptor mechanisms are very different between insects and mammals, and also involved in development in insects. Studying the diversity in this pathway across organisms suggests treatments for human infections with antimicrobial proteins from different sources.

1.3.2 Driving Biological Questions

The above case studies on two different pathways ask some common questions. This thesis is driven by these questions that revisit broad evolutionary principles in the new light of cellular pathways. The specific contributions of the thesis are outlined in Section 1.4.

- What similarities and differences (variation) exist in a pathway present in many species? Heritable variation could exist at the level of pathway gene contents, in-

teraction patterns, molecular sequences of pathway components, their biochemical functions, etc.

- How did variation arise in a pathway? Evolutionary events that resulted in the present-day variants of the pathway could be recruitment or loss of genes or interactions in the pathway due to mutations in coding or regulatory sequences, horizontal gene transfer, etc.
- What is the interplay of evolutionary forces acting on the components of a pathway? Little is known about how the forces (e.g. random drift, natural selection) on the different genes in a pathway work in concert to adapt the whole pathway to new environmental niches.

Seeking a unified approach to answer these questions for any pathway of interest is a fundamental challenge that could engage researchers for decades. For this purpose, we seek common methodology and principles to systematically study the evolution of any particular pathway. Attempts at conceptual unification in biology are provocative as it is not clear *a priori* if a unification is possible or even useful to further our understanding of cellular phenomena compared to specific case studies (Lenski *et al.*, 2006).

However the evolutionary theory we have for sequences (e.g. neutral mutation theory, which assigns a greater role to random genetic drift than natural selection in explaining sequence changes across species) and its wide-ranging impact (e.g. neutral theory predicts that functional regions of a genome evolve slower than other regions, which forms the central premise of comparative genomics), encourages efforts to explore an evolutionary theory for pathways. Initial efforts to systematically study pathway evolution are promising when focused on a class of pathways such as metabolic pathways involved in amino acid biosynthesis (Forst and Schulten, 2001), or pathways that repair abnormal DNA structures (e.g. chemically modified bases or base-pairing mismatches) (Eisen and Hanawalt, 1999). These attempts clarify the challenges in extending and complementing the useful formalisms on sequence evolution of individual molecules to explain the concerted evolution of molecules in a pathway.

1.3.3 Challenges in Computational Methods

Computational methods (e.g. alignment algorithms) and mathematical models (e.g. phylogenies) provide a formal language to express and study evolutionary concepts. As an example, sequence alignment explicitly characterizes the variation in a set of homologous sequences and phylogeny of the sequences captures the evolutionary events that best explain this variation. The interrelated problems of alignment and phylogeny fall under the umbrella of comparative and evolutionary methods respectively.

Research on comparative and evolutionary methods in the context of pathways is only emerging. The rich network structure of a pathway makes it both interesting and hard to develop such methods. Let me illustrate with a comparative and an evolutionary example. (a) Alignment of one-dimensional sequences is efficiently solved by dynamic programming, but several problem formulations in the alignment of large networks (e.g. cell-wide network of protein interactions) are intractable (NP-hard (Garey and Johnson, 1979)). (b) Correlated evolution of even two interacting proteins (e.g. a ligand and its receptor) is difficult to model probabilistically at a fine-grained level due to a large number of associated parameters. Hence to study how all the components of a pathway evolve in a correlated fashion, we need research on co-evolution models with simplifying abstractions and assumptions.

Rigorous computational methods are preferable over ad-hoc ones to obtain reliable insights on conservation and evolution of cellular pathways. The emphasis is on the design of methods with provable guarantees on their correctness and running time wherever possible, and principled approximations or heuristics for intractable problems.

1.4 Thesis Contributions

Comparative and evolutionary analysis of biomolecular pathways involves significantly extending the concepts, formalisms and methods on the alignment and phylogeny of sequences to networks and pathways. This conceptual and computational exercise provides substantial opportunities and challenges. The primary contribution of this thesis is the

design of new and rigorous computational methods to study the conservation and evolution of cellular pathways. In more specific terms, our contributions are:

- An original network comparison method to identify protein modules conserved between the protein interaction networks of two different species.
  - Unlike previous methods, our method is based on a provably efficient algorithm that we designed to identify matching subgraphs between two graphs. This algorithm with its quite general framework is novel in the broader field of graph-matching as well.
  - Our network comparison method performs competitively against previous methods in comparisons of the yeast protein network with the human, fruit fly and nematode worm networks (evaluated using known yeast protein complexes), and in prediction of protein functions at the pathway level.
- A novel probabilistic approach to estimate the phylogeny of a pathway present in closely related species. Numerous studies document the variation in a pathway across species, and our approach is one of the very few that explain this variation along a phylogenetic tree.
  - Our estimation is based on a probabilistic model that makes its assumptions about the evolution of a pathway explicit, unlike previous discrete pathway similarity based approaches. The model we design captures the correlated fashion in which the pathway genes are gained or lost over evolutionary time, by using pathway gene content and interaction data. The model represents this data using a succinct, probabilistic network to achieve a tractable number of parameters.
  - The estimated phylogenetic trees of certain bacterial and archaeal pathways, when compared with known species phylogenies and data on cellular phenotypes, lead to interesting hypotheses about the evolution of such pathways as citric acid cycle or chemotaxis.

Chapter 2

Research Overview

We take a computational approach to inquire about the conserved nature and evolutionary history of cellular pathways. The approach thus primarily involves the formulation of a computational problem to address a concrete question about pathway conservation or evolution, design of a rigorous computational method to solve the problem, and application of the new method on relevant genomic data to generate biological insights. The thesis presents two studies that follow this thread of research. The first study is on identifying pathway structures conserved across protein interaction networks of different species, and the second is on reconstructing the evolution of a pathway present in closely related species [1]. This chapter provides a context, related work and overview for these studies, and later chapters will provide a detailed description of the computational problems, methods and results arising in these studies.

2.1 Protein Network Comparison

Biological sequence comparison and alignment is a well-established research area in computational biology. A very recent and emerging research area stimulated by large-scale experimental data on cell-wide interactions such as protein-protein interactions (Bork

[1]The first study is joint work with Richard M. Karp, and the second study is joint work with Amoolya H. Singh and Richard M. Karp. Please see Acknowledgements.

et al., 2004) is biological network comparison. Research in this area involves the extension of concepts and methods from comparison of sequences to comparison of biological networks. For instance, we can compare the protein interaction networks of two species to detect conserved pathway structures between them, as we do in our research study. An important application of detecting such conserved pathways is the identification of functional modules of proteins from raw protein interaction data. The overview of our study begins with this application below.

2.1.1 Identification of Functional Modules

A comprehensive map of the pathways in a cell is a useful starting point for studies that place an unknown gene in the context of its pathways, hypothesize novel components of and functional links between pathways (even well-studied ones), find the pathways involved in a complex phenotype such as a disease, or infer the evolution of pathways whose data is available across species. One promising approach to cataloging and exploring cellular pathways is to organize the myriad protein interactions observed in a cell into functional modules of proteins (Hartwell *et al.*, 1999). The need for such organization is readily evident from a glance at the network of interactions observed in the yeast *Saccharomyces cerevisiae* (Figure 2.1).

Large-scale data on protein interactions are publicly available for many model organisms and humans from high-throughput though noisy experiments or extensive literature scanning (Bork *et al.*, 2004). Examples include large protein interaction datasets for the yeast *Saccharomyces cerevisiae* (Uetz *et al.*, 2000; Ito *et al.*, 2001; Ho *et al.*, 2002; Gavin *et al.*, 2002, 2006; Krogan *et al.*, 2006), the fruit fly *Drosophila melanogaster* (Giot *et al.*, 2003), the nematode worm *Caenorhabditis elegans* (Li *et al.*, 2004), and humans (Peri *et al.*, 2003; Rual *et al.*, 2005; Stelzl *et al.*, 2005). There are also efforts to consolidate and curate both large-scale and small-scale molecular interaction data published for different organisms (for instance, refer the International Molecular Exchange consortium of public interaction data providers at `http://imex.sf.net/`).

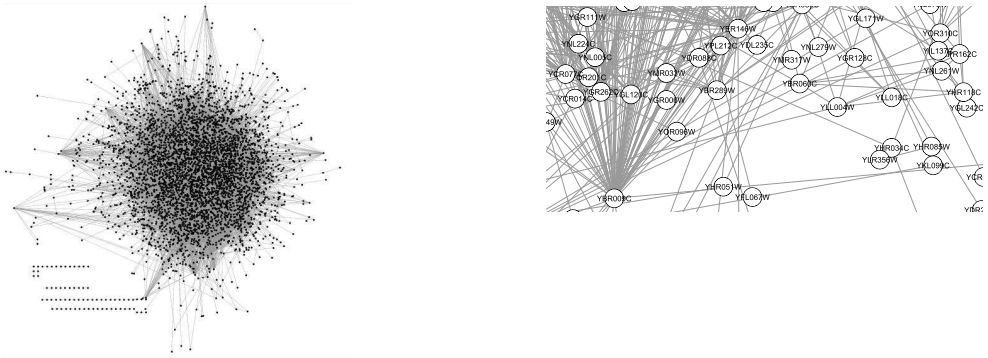

Figure 2.1. Global view of the protein interaction network of *Saccharomyces cerevisiae* (left) comprising 14,319 interactions over 4389 proteins, and a zoomed view of some proteins (right). A dot or circle represents a protein and a line joins two interacting proteins. The protein networks in this thesis are drawn using the Cytoscape software (Shannon *et al.*, 2003).

The quality of the above datasets on physical interactions between proteins, in terms of coverage of cell-wide interactions and accuracy of detected interactions, depends on the scale and quality of their experimental sources (von Mering *et al.*, 2002; Deng *et al.*, 2003). The sources of the datasets range from high-throughput experiments based on yeast two-hybrid methodology (Uetz *et al.*, 2000; Ito *et al.*, 2001; Giot *et al.*, 2003; Li *et al.*, 2004; Rual *et al.*, 2005; Stelzl *et al.*, 2005) or affinity purification and mass spectrometry methodology (Ho *et al.*, 2002; Gavin *et al.*, 2002, 2006; Krogan *et al.*, 2006), to extensive literature scanning for interactions detected in small-scale experiments as well (Peri *et al.*, 2003). In the yeast two-hybrid method, two proteins are tested for interaction by fusing one protein to the DNA binding domain and another to the activation domain of a split transcription factor of some reporter gene. If the proteins physically interact, then expressing the fused proteins in the nucleus of a yeast cell results in a functional transcription factor of the reporter gene. The method hence detects stable as well as transient binary interactions *in vivo*, but it also results in false positives as the proteins are not tested in their native conditions (e.g. cellular compartments). Affinity purification and mass spectrometry methodology can detect multi-

protein complexes in their native conditions. The affinity purification step captures and purifies a complex of proteins, if any, that a tagged "bait" protein participates in, and the mass-spectrometry step detects the components of the purified complex. The method, depending on the specific protocol used, can be accurate in detecting stable complexes (Deng *et al.*, 2003), but it misses transient interactions that get lost during purification.

Computational methods are valuable to interpret the networks of these raw protein interactions of different organisms and cope with their scale. Methods that organize protein networks into functional modules are similar in spirit to ones that organize raw genomic sequences into functional elements like genes, regulatory sequences, etc. Before giving an overview of these methods, we clarify certain terms. In this thesis, a *protein (interaction) network* refers to a graph whose nodes are the proteins of an organism and edges indicate physical interactions between proteins inferred from experiments or literature as discussed above. A *protein module* refers to a subset of proteins in this network along with the induced interactions between them, and a *functional module* is then a protein module known or supposed to be involved in a common cellular pathway (e.g. a protein complex involved in import of proteins into mitochondria).

2.1.2 Previous Work

We provide a background on methods that find functional modules in protein networks. We review single-species methods in brief as they are not our main focus, and cross-species, network comparison methods in detail. We also discuss concepts in graph-matching and data-mining that are related to network comparison.

Single-species methods. Several methods analyse a single organism's protein network to identify functional modules. A typical single-species method uses connectivity information to cluster a protein network into highly connected modules, and known functional annotations of proteins to interpret or validate the resulting modules (see review (Bork *et al.*, 2004)). Examples range from early methods applied on various yeast protein network data (Bader and Hogue, 2003; Rives and Galitski, 2003; Spirin and Mirny, 2003; King *et al.*,

2004; Pereira-Leal *et al.*, 2004) to some very recent ones (Hwang *et al.*, 2006; Pu *et al.*, 2007). These and many other published methods differ in the clustering algorithms they employ, whether they integrate additional information such as gene expression or functional data in the clustering, and the species-specific networks they study (see detailed review (Sharan *et al.*, 2007)).

A single-species clustering method is effective if the modules it outputs align well with a set of reference or known functional modules, and if the output modules that don't match any known functional module suggest plausible and testable biological hypotheses. Despite active research on single-species methods for the past few years, systematic comparison of the effectiveness of available methods is lacking. Welcome exceptions are two recent studies (Brohée and van Helden, 2006; Hwang *et al.*, 2006) that compare a subset of the available methods.

Cross-species methods. A few recent methods compare protein networks from two or more species to identify functionally similar (conserved) protein modules between them (see review (Sharan and Ideker, 2006)). These pairwise or multiple network comparison methods improve over single-species methods by using information on conservation (cross-species similarity of protein sequences and interaction patterns) as well as connectivity of the networks. For instance amidst noisy data, conservation could reinforce evidence that some connected proteins participate in a common function. These comparative methods also enable transfer of functional annotations between organisms at the level of conserved modules, interactions and proteins, a key utility single-species methods cannot provide. Similarly, conserved modules could provide a basis for studies on the evolution of cellular structures and networks.

Current network comparison methods include NetworkBLAST (Sharan *et al.*, 2005), MaWISh (Koyuturk *et al.*, 2005), a Bayesian alignment method (Berg and Lassig, 2006), and Græmlin (Flannick *et al.*, 2006). At a high level, these methods formulate a biologically-inspired measure to score when a set of subgraphs from the input networks constitute a conserved module and use heuristics to search for all high-scoring similar subgraphs between

the networks. Search heuristics are necessary as the scoring measures are complicated and lead to intractable (NP-hard (Garey and Johnson, 1979)) search problems. We give a brief description of each method next to clarify the scoring measures.

NetworkBLAST constructs an alignment graph from the multiple input networks (limited to two or few networks to constrain the alignment graph size), weighs its nodes and edges according to a scoring measure that decomposes into node and edge components, and searches it for high-scoring subgraphs. A node of the alignment graph represents a set of evolutionarily related (homologous) proteins across the input species, an edge between two nodes denotes a conserved interaction, and node and edge weights are from a log likelihood ratio score. The likelihood ratio is under a probabilistic model that favours detection of dense modules of sequence-similar proteins against a random background. The authors of NetworkBLAST draw these concepts from their earlier works on finding conserved linear paths (Kelley *et al.*, 2003) and dense clusters (Sharan *et al.*, 2004) between a yeast and a bacterial protein network. MaWISh aligns two networks using similar concepts but their node and edge scores are based on an empirical model of network evolution through events such as gene duplication and interaction gain/loss. In detail, MaWISh uses a duplication/divergence model where a duplicated gene inherits interactions to all neighbors of the original gene, and these interactions later diverge (are gained/lost) over evolutionary time. This model inspires a scoring measure between two aligned subnetworks that is derived from matched and mismatched interactions between orthologous (or paralogous) protein pairs. The Bayesian alignment method compares two networks using a probabilistic model of network evolution that integrates node and edge evolution components in different proportions. Model parameters and high-scoring alignments are inferred from a Bayesian analysis. Græmlin progressively aligns multiple networks using node scores based on phylogenetic history of proteins and edge scores suited to detect conserved modules of different topologies.

Probabilistic models of network evolution are becoming popular in the design of network comparison scoring measures. Very recent pairwise (Hirsh and Sharan, 2007) and multiple (Dutkowski and Tiuryn, 2007) network comparison methods are examples. The first

method is a close successor of NetworkBLAST and uses a model where a conserved ancestral complex evolves in a single step to the observed interactions in the input networks. The second method uses a similar notion of conserved ancestral network, but a more detailed probabilistic model that traces interaction gain/loss at every speciation or gene duplication event. These events are inferred from standard phylogenetic analysis of proteins, and used in the rest of the analysis to obtain a tractable probabilistic model and search procedure for multiple input networks. To summarize, each method's scoring measure makes many direct or indirect assumptions about the underlying network evolution and impacts the computational complexity of the search procedure. More research is clearly needed on the design and evaluation of both scoring measures and search problems to reach a consensus on the appropriate method to use.

Other network comparison methods. Precursors of protein network comparison methods are studies on cross-species conservation at the level of protein interactions ("interologs" introduced in (Walhout *et al.*, 2000)) rather than modules. An application of interologs is the use of known protein interactions in one species to predict novel interactions between homologous proteins in another species (e.g. (Matthews *et al.*, 2001)). Concepts similar to interologs include conserved protein-DNA interactions or "regulogs" (e.g. (Yu *et al.*, 2004)), and conserved coexpression relationship between gene pairs (e.g. (Stuart *et al.*, 2003)).

Research in network comparison extends beyond protein networks to other types of biological networks such as networks of metabolic reactions. A study (Ogata *et al.*, 2000), which predates protein network comparison, compares the metabolic network of a species with a linear network that reflects the genome order of enzyme-coding genes in the species. This study heuristically finds modules of enzymes that catalyze contiguous metabolic reactions and cluster along the genome, by single-link clustering a product graph of the two input graphs. Metabolic networks across multiple species were compared in (Chor and Tuller, 2006) to find conserved modules and pairwise distances using a notion of relative description length (length of describing one graph, given the description of another). A recent

method SAGA (Tian *et al.*, 2007) searches for approximate occurrences of a query pathway in a large dataset of metabolic and other biological pathways. The method indexes the dataset pathways to accelerate search of small and sparse query pathways. Index structures are popular among query-dataset graph searching methods in the pattern-matching and data-mining fields (see survey (Shasha *et al.*, 2002) and references in (Tian *et al.*, 2007)). Another idea inspired from data-mining is the mining for frequently occurring modules in metabolic or protein networks (Koyuturk *et al.*, 2006).

Comparative analysis of protein networks with networks of genetic interactions (functional association between gene pairs whose double mutant forms show reduced fitness or lethality) (Kelley and Ideker, 2005) or gene regulatory interactions (interactions between transcription factors and the target genes they regulate) (Tan *et al.*, 2007) is possible too. Researchers are actively designing and applying comparative methods to extract biological insights from the deluge of network data.

Graph-matching. The computational problems in these network comparison methods have close connections to the broader field of graph-matching. Graph-matching refers to a class of problems that find similar subgraphs between two graphs (see (Schellewald, 2005) for other related classes). Many graph-matching problems in the literature are NP-hard (Garey and Johnson, 1979), permitting only heuristic or approximate solutions, due to a stringent global structural match that they require between the similar subgraphs. For instance, the maximum common subgraph problem requires an exact isomorphic match between subgraphs of the two input graphs and is NP-hard (Garey and Johnson, 1979). Finding the occurrence of an entire pattern graph H in another graph G is well-known to be NP-hard too under the subgraph isomorphism or homeomorphism formulations (Garey and Johnson, 1979). Subgraph isomorphism demands a one-to-one mapping of all nodes in H to some nodes in G such that the edges in H map to the corresponding edges in G. Subgraph homeomorphism is more general in that the edges in H need only map to edge/node-disjoint paths between the corresponding endpoints in G. Certain restricted variants of these problems when H, G are trees permit polynomial-time algorithms (see

(Pinter *et al.*, 2004) and references therein) and have been used to align tree-like metabolic pathways (Pinter *et al.*, 2005).

Problems that require inexact match between graphs to deal with error-prone data are NP-hard too for a family of graph edit cost functions (Bunke, 1999). Some studies use a scoring function to measure how similar two subgraphs are. They find high-scoring subgraph pairs by finding heavy-weight subgraphs in a product graph of the input graphs (e.g. alignment graphs in (Sharan *et al.*, 2005; Koyuturk *et al.*, 2005)), which is also NP-hard by reduction from the maximum weight induced subgraph (Koyuturk *et al.*, 2005) or maximum clique problem (Garey and Johnson, 1979). However, if we restrict to finding heavy-weight simple paths of a fixed length in a graph, then fixed-parameter tractable algorithms are possible based on the idea of random acyclic orientations or color-coding (Alon *et al.*, 1995). These algorithms have been applied to protein network comparison in (Kelley *et al.*, 2003) and extended as well in (Scott *et al.*, 2005). Fixed-parameter tractable algorithms are efficient only for small values of the parameter though, which in our case is the length of the heavy-weight paths. Efficient algorithms exist for longer paths if we further restrict the problem by fixing a given query path and finding its approximate, possibly repeated, occurrences in a larger graph (Yang and Sze, 2007).

2.1.3 Our Graph-matching Algorithm

We present a pairwise network comparison method based on a graph-matching algorithm with provable guarantees. Our search formulation is applicable for comparing two general graphs (representing protein networks of two species) to find conserved protein modules. Our novel graph-matching algorithm and the guarantees on both its correctness and running time make this work markedly different from previous methods relying on search heuristics. The search formulation is also biologically meaningful and yields promising results in detecting functional modules and transferring functional annotations.

In more detail, we formulate a conserved protein module as a pair of connected and locally matching subsets of proteins, one from each input network. By a locally matching

subset pair, we mean that every protein in one subset is similar to some protein in the other subset, at the level of both their sequences and neighborhood or context in the networks. Search for such conserved modules is simpler than search formulations in previous network comparison methods and hence admits an efficient polynomial-time algorithm. This polynomial-time search problem is novel in the broader field of graph-matching as well. The main operation of this algorithm is a recursive match and split of the proteins in the two input networks. We assess the statistical significance of the conserved modules found by the algorithm, based on a similarity score of the conserved module and estimates of noise in the interaction data.

We apply our method to compare the yeast protein network with the human, fruit fly and nematode worm protein networks. We evaluate the detected conserved modules using known yeast protein complexes and compare its performance to previous network comparison and single species clustering methods. We also demonstrate the utility of network comparison for predicting pathway annotations of human proteins and validate the predicted annotations. The results suggest that our method is a promising, provably efficient alternative to current protein network comparison methods. Further, our algorithm framework is quite general and hence has applications in comparing biological networks beyond protein networks (once a relevant matching criterion is chosen).

2.2 Pathway Phylogeny Estimation

Numerous studies document the variation in the components (genes, interactions, etc.,) of a pathway present in multiple species, but very few attempt to explain this variation in terms of events in the evolutionary history of the pathway. Our study seeks to explain the evolutionary history of the extant variants of a pathway using the concept of phylogenetic trees. We present models and methods to build phylogenetic trees of any pathway of interest, under the premise that the pathway is evolving as a heritable unit across closely related species.

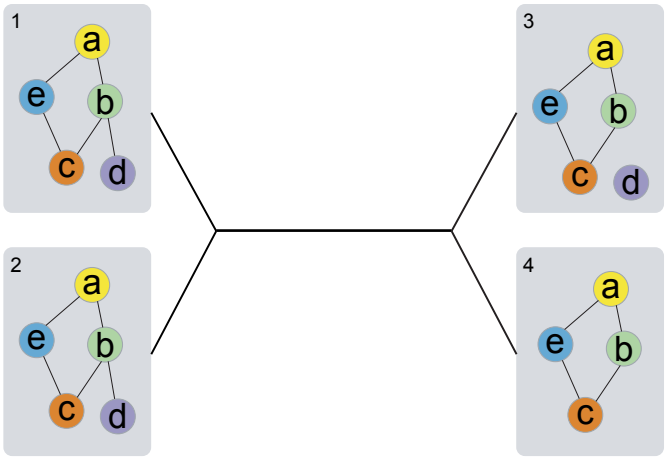

Figure 2.2. The phylogeny of a pathway of genes a, b, c, d, e in four species, numbered 1-4 and outlined in gray. The interaction between b, d is lost in Species 3 and gene d is lost in Species 4, probably because gene d is not very important for the function involved.

2.2.1 Context for the Phylogeny of a Pathway

Evolutionary studies that use sequences to determine common ancestry assume that the gene is the unit on which selection acts, and that differences in sequence could explain differences in evolutionary history (Zuckerkandl and Pauling, 1965; Batzoglou, 2005). Recent work, however, suggests that selection might be acting at higher levels of cellular organization than individual genes; that is, at the level of biochemical networks or regulatory pathways (Boldogkoi, 2004). To examine this hypothesis in a systematic way, we need formal evolutionary models and estimation methods to reconstruct the phylogenetic history of a pathway, analogous to those we have for a gene.

Researchers have hence focused on formally building phylogenetic trees of pathways that predict how a pathway evolved from its ancestral form(s) to its present-day form(s) in different species (Figure 2.2). One existing approach aggregates pairwise sequence distances between member genes in a pathway to obtain pathway distances and uses it to obtain the phylogeny of electron-transfer and amino acid biosynthesis pathways (Forst and Schulten, 2001). Another approach derives an iterative, graph similarity metric between two path-

ways based on the nodes (enzymes) of the pathways and the graph structural relationship between the nodes, and uses it with a distance matrix method to build phylogenies for several respiratory and carbohydrate metabolism pathways (Heymans and Singh, 2003). A recent distance matrix approach uses a similarity metric based on relative description length between two graphs, and builds the phylogeny of a species set from the repertoire of all metabolic pathways observed in these species (Chor and Tuller, 2006). Conceptually, this method could also build the phylogeny of a particular metabolic pathway whose variants are present in multiple species.

These approaches compute distances between pathways using graph-theoretic or combinatorial similarity measures. However, probabilistic models are often preferred over similarity-based methods to estimate phylogenetic distances, as they make explicit any assumptions about pathway evolution and lead to consistent estimates (Felsenstein, 2003; Durbin *et al.*, 1998). But detailed probabilistic models of pathway evolution that try to explain low-level mechanisms of evolution may suffer from other problems, the commonest of which is overparameterization. For example, consider the evolution of a pathway of just two genes, a ligand and its receptor. A perfectly realistic model should account for how nucleotide pairs in the two genes are evolving in a correlated fashion (Pollock *et al.*, 1999). To do so, however, the model would need to consider a number of correlations quadratic in sequence length, since it is not known which nucleotide pairs are co-evolving. This ultimately leads to too many parameters and reduces the ability of the model to predict the true phylogeny (Sullivan and Joyce, 2005).

2.2.2 Our Unified Phylogenetic Model

In our study, we present a probabilistic model for the evolution of a pathway that attempts to address the dual requirements of tractability and biological accuracy. The model represents a pathway by the presence or absence of genes (characters) in different species, and uses information about interactions between pathway genes to model the dependency between gene gains and losses. The assumptions behind these choices are consistent with our premise that the entire pathway is evolving as a heritable unit. The model can be used

with a maximum likelihood (ML) or distance matrix method to formally build pathway phylogenies.

More specifically, we propose a phylogenetic model to explain the variation in the gene content of a pathway present in closely related species, assuming a static dependence structure of gene gains and losses dictated by canonical pathway interactions. A canonical view of the pathway interactions is derived from a knowledge of the pathway interactions observed in model organisms. Our focus on closely related species allows us to take this canonical view and leads to a tractable model, as we may assume that the pathways haven't diverged too much through gene duplication or other radical mutation events. Our model is a special case of a co-evolution model of k binary characters that we developed by extending a previously published 2-character co-evolution model (Barker and Pagel, 2005; Pollock *et al.*, 1999). Whereas a vanilla extension requires a number of parameters exponential in k, our model has a tractable number of parameters due to the use of a Markov network (Jordan, 1999) formulation derived from the canonical pathway interactions and an assumption about uniform evolutionary rate for characters. Our model is limited to $k \leq 13$ in our experiments due to computational constraints such as memory size. We note that research on co-evolution models over larger number of characters is active and open (e.g. (Pedersen and Jensen, 2001)).

We apply the model to estimate the phylogenies of several pathways present in bacteria and archaea, for which there is a wealth of genetic, genomic and phenotypic data available. We study essential pathways such as glycolysis and the citric acid cycle, stress response pathways such as chemotaxis, and cell-cell communication pathways such as quorum sensing. We suggest broad hypotheses about the evolution of these pathways across bacterial and archaeal species, by comparing their estimated phylogenies with known species phylogenies and correlating any discrepancies found with data on cellular phenotypes. Our results suggest that a systematic approach to build and interpret pathway phylogenies is a useful step towards mapping the relationship between phenotype (pathway function) and genotype (genetic content), a basic task in evolutionary biology.

Chapter 3

Graph Comparison of Protein Interaction Networks

We study the problem of comparing the protein interaction networks of two species to detect functionally similar (conserved) protein modules between them. We motivated this problem and highlighted our contributions in the context of earlier works in previous chapters. In this chapter, we fully describe our pairwise network comparison method and results from its application to compare the protein networks of different species. As mentioned, our method is based on an algorithm we developed to identify matching subgraphs between two graphs, and unlike previous network comparison methods, our algorithm has provable guarantees on correctness and efficiency. Our algorithm framework also admits quite general criteria that define when two subgraphs match and constitute a conserved module. Further, the results we obtain pertaining to evaluation of the detected conserved modules, and their associated functional descriptions and predictions are competitive relative to previous methods.

3.1 Methods

Recall that a protein (interaction) network refers to a graph whose nodes are the proteins of an organism and edges indicate physical interactions between proteins (see also Figure 2.1), and a protein module refers to a subset of proteins in this network along with the induced interactions between them.

3.1.1 Conserved Module Premise

Our method compares protein networks from two species to find functionally similar or conserved protein modules between them. A conserved module is intuitively a pair of protein modules that share cross-species similarity at the node level (homology or evolutionary relationship of corresponding protein sequences) and graph structure level (pattern of interactions). Our method's specific *premise* is that a conserved module is a pair of *connected* and *locally matching* subsets of proteins, one from each input graph. Two proteins locally match if their sequences and neighborhood in the network are similar, and a collection of such protein pairs is a locally matching subset pair. We support this premise in this section and formalize it in the next.

Our premise is biologically motivated. The premise's connectivity criterion makes it likely for a subset of proteins to be functionally homogeneous and its local matching criterion makes it likely for two protein subsets to be functionally similar. Moreover, these criteria are minimal requirements and hence sensitive in detecting reference functional modules. To illustrate, a functional module's counterparts in two different species needn't match exactly in their graph structure due to evolutionary divergence or errors in the interaction data, which makes a local match more sensitive than a stringent structural match like exact isomorphism. The two criteria yield good specificity too with some additional improvements detailed in Section 3.1.3. Note that a method's sensitivity denotes the fraction of reference modules it detects and specificity the fraction of modules output by it that match some reference module.

Our premise is also computationally attractive as it results in a search problem that

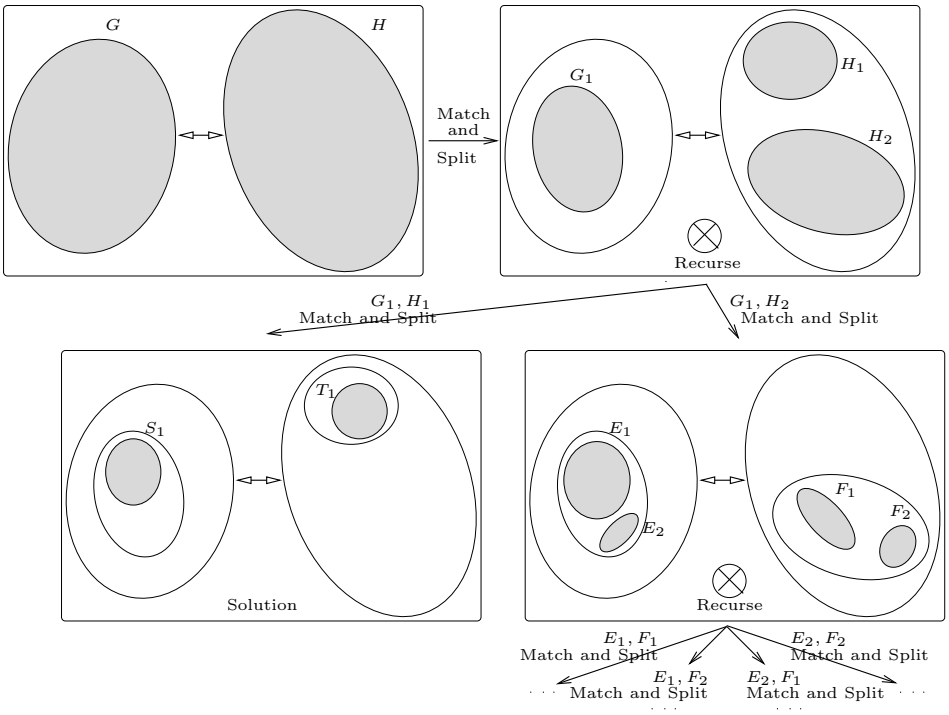

Figure 3.1. Pictorial sketch of the main operations of our graph-matching algorithm. We show the input graphs G, H and their subgraphs as ovals, hiding node and edge details. The algorithm focuses only on the shaded subgraph pairs at any point in its execution, and refines them recursively until all solutions (similar subgraph pairs) are found. A refinement involves doing a match and a split step to compute locally matching and connected node-sets between two shaded subgraphs (see text for details). The subgraph pair S_1, T_1 is a solution, and the algorithm might find more solutions as it recurses on the subgraph pairs E_i, F_j. The statistically significant solutions are finally output as conserved modules.

admits provably good algorithms for many choices of the connectivity and local matching criteria. As mentioned in the previous chapter, our method's tractable search formulation distinguishes it from previous network comparison methods and graph-matching problems.

3.1.2 Graph-matching Engine

Finding conserved modules under the premise just outlined reduces to a graph-matching problem of finding similar subgraphs between two input graphs. We present this graph-matching problem variant and our polynomial-time algorithm for it in this section. We also discuss the algorithm's generality, which makes it relevant for application areas beyond protein networks.

As a prelude, we sketch our graph-matching engine in Figure 3.1 and informally outline it now in the context of protein networks. Say we start with the yeast and human protein networks along with the homologous protein pairs between them. Our algorithm first computes *locally matching* proteins (i.e., proteins with similar sequences and local context) between the two networks, and safely discards the other proteins (i.e., any yeast protein with no human homolog or with some human homolog but with poor local match, and vice versa). The algorithm next splits the remaining yeast and human networks into *connected* sets of proteins. These connected subgraphs are locally matching with respect to the full input networks. Our algorithm then repeats the above match and split steps on each pair of the connected subgraphs recursively, until the final similar subgraph pairs are found (as in Figure 3.1). The ensuing text provides precise descriptions on matching general graphs.

Problem statement

We are given as input two graphs and a node similarity function $sim(\cdot, \cdot)$. The function $sim(u, v)$ is true whenever node u is similar to node v (e.g. based on sequence similarity of proteins) and false otherwise. This $sim(\cdot, \cdot)$ is a symmetric function defined over all pairs of nodes u, v, one from each input graph. The problem now is to list pairs of connected and locally matching subgraphs between the input graphs. In this work, a subgraph of a graph usually refers to an induced subgraph, which is a subset of nodes in the graph along with all edges between them.

We first build a *local-match*$_{S,T}(u, v)$ function for any subgraph pair S, T of the input graphs using the $sim(\cdot, \cdot)$ function. This new function captures local or contextual match between the nodes u of S and v of T using similar local structures present around these nodes in S and T. Two possible choices of similar local structures are: *similar length-p paths* for some small p (say 2), and *s-similar neighborhoods* around nodes for some small s (see Figure 3.2). In the former case, *local-match*$_{S,T}(u, v)$ is true whenever some length-p path in S containing u is similar to some length-p path in T containing v (two paths are similar if all their corresponding nodes are similar according to $sim(\cdot, \cdot)$). In the latter case, *local-match*$_{S,T}(u, v)$ is true whenever $sim(u, v)$ is true and $sim(u', v')$ is true for at least

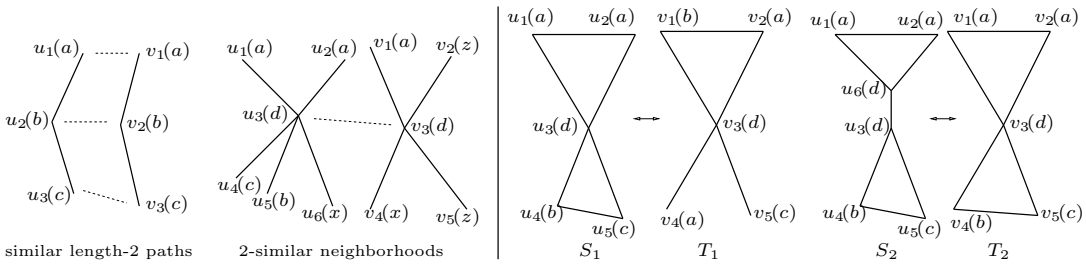

Figure 3.2. Illustration of two similar local structures (left) and two solutions (right). Assume $sim(u_i, v_j)$ is true whenever u_i, v_j have the same label shown inside brackets. The dotted lines (left) show some of the locally matching node pairs. For instance, u_3 locally matches v_3 based on 2-similar neighborhoods since they share label d and two of their neighbor pairs u_1, v_1 and u_6, v_4 share labels a and x respectively. The subgraph pair S_1, T_1 is a solution when the local matching criterion is based on similar length-p paths (p=1 or 2) or 1-similar neighborhoods (i.e., when the criterion is 1-similar neighborhoods say, every node in S_1 locally matches some node in T_1 and vice versa). The subgraph pair S_2, T_2 is a solution when the criterion is based also on 2-similar neighborhoods.

s distinct neighbor pairs u' of u in S and v' of v in T. The stringency of this criterion increases with s, with 1-similar neighborhoods being the least stringent. Properties of a *local-match*$_{S,T}(\cdot,\cdot)$ function such as polynomial-time computability and monotonicity, which yield tractable problem formulations, are discussed later along with the algorithm.

We are ready to state the problem. Given two input graphs G, H, a node similarity function $sim(\cdot,\cdot)$, and a *local-match*$_{S,T}(\cdot,\cdot)$ function computable for any two subgraphs S, T, the problem is to find all maximal induced subgraph pairs $S \subseteq G$, $T \subseteq H$ that satisfy two criteria:

Connectivity: S, T are each connected, and

Local Matching: Each node u in S locally matches at least one node v in T according to the *local-match*$_{S,T}(u, v)$ function, and vice versa.

Any subgraph pair that satisfy the above two criteria is called a *solution* (see Figure 3.2), and maximality requires that of two solutions S, T and S', T' with $S' \subseteq S, T' \subseteq T$, we only output the maximal one S, T to avoid redundancy.

Algorithm and guarantees

We present a simple and efficient algorithm for the above problem for any *monotone* local matching criterion. A monotone criterion is one where any two nodes that locally match remain so even after adding more nodes to the subgraphs under consideration (i.e., if *local-match*$_{S,T}(u, v)$ is true, then *local-match*$_{S',T'}(u, v)$ is also true for any $S' \supseteq S, T' \supseteq T$). The similar length-$p$ paths or s-similar neighborhoods criterion from last section are monotone. A useful property of monotonicity is that the maximal solutions are only quadratic in number, since it lets us merge any two solutions S, T and S', T' with a common node pair u, v (i.e., $u \in S \cap S', v \in T \cap T'$) into one solution $S \cup S', T \cup T'$.

Our algorithm presented below matches and splits the nodes of G and H into smaller components, and then recurses on each of the component pairs. For induced subgraphs $S \subseteq G, T \subseteq H$, we let $lm(S, T)$ denote all nodes u in S for which *local-match*$_{S,T}(u, v)$ is true for some node v in T.

Match-and-Split(G, H):
[Match] Compute induced subgraph:

G' of G over the locally matching nodes $lm(G, H)$, and

H' of H over the locally matching nodes $lm(H, G)$.

[Split] Find connected components:

$G_1, \ldots, G_c$ of G', and

$H_1, \ldots, H_d$ of H'.

[Recurse]
if ($c = 1, d = 1$ and $G' = G, H' = H$)

Output the maximal solution G, H. [base case]

else

for $i = 1$ to $c, j = 1$ to d

Match-and-Split(G_i, H_j). [recursive case]

Correctness. Consider any solution S, T. Each recursive call retains all locally matching nodes and processes all pairs of resulting components. Hence in at least one path of the recursion call tree, all nodes in S, T remain locally matching due to monotonicity and connected as part of a bigger subgraph pair. This retained unsplit S, T is finally output as part of a solution. Further, the output solutions are maximal because no node pair is common to any two output solutions (as shown below in the proof of Lemma 3).

Running time. Let n_F, m_F denote the number of nodes, edges respectively in a graph F. The algorithm runs in time $O(n_G n_H + (n_G + n_H) m_G m_H)$ on the graphs G, H when the local matching criterion is similar length-1 paths. The algorithm is efficient in practice too as the locally matching proteins between two graphs in our experiments reduces drastically as the recursion depth increases.

Analysis of running time

The running time bound mentioned above follows from bounds proved here, which hold for more general monotone local matching criteria. Let n_F, m_F be as defined above. Let $f(G, H)$ be a function bounding the process time (of the non-recursive match and split steps) on the graphs G, H, and f_0 be the constant term in $f(G, H)$ (technically $f(G, H) \doteq f(n_G, m_G, n_H, m_H)$ and $f_0 \doteq f(0, 0, 0, 0)$).

Theorem 1 *The Match-and-Split(G, H) algorithm's running time when using a monotone local matching criterion is bounded by*

[general] $O(n_G n_H f(G, H))$ *for a general* $f(\cdot, \cdot)$*, and*

[special] $O(n_G n_H + (n_G + n_H) f(G, H))$ *for a class of* $f(\cdot, \cdot)$ *that satisfies the condition* $\sum_{i=1}^{c} \sum_{j=1}^{d} (f(G_i, H_j) - f_0) \leq (f(G, H) - f_0)$ *for any* G, H*.*

From the simple bound that holds for a general $f(\cdot, \cdot)$, we see that the algorithm runs in polynomial time if $f(\cdot, \cdot)$ is asymptotically bounded by a polynomial. The better running time bound applies for a broad class of $f(\cdot, \cdot)$, since the special condition above holds for many reasonable functions including any polynomial $f(\cdot, \cdot)$ whose monomials

each contain the factors n_G or m_G, and n_H or m_H. For such polynomials, the condition follows readily from the connected components of a graph being node-disjoint and edge-disjoint. To illustrate, let the local matching criterion be similar length-1 paths. Then $f(G,H) = k\, m_G m_H$ for some constant k as the match step simply considers all edge pairs in G, H and the split step runs a linear-time connected components procedure. This $f(\cdot,\cdot)$ satisfies $\sum_{i=1}^{c}\sum_{j=1}^{d} k\, m_{G_i} m_{H_j} = k\, \sum_i m_{G_i} \sum_j m_{H_j} \leq k\, m_G m_H$, where the crucial last step is from the disjointness of the G_i's and the H_j's. Other criteria like similar length-p paths or s-similar neighborhoods also yield a polynomial $f(\cdot,\cdot)$ satisfying the condition.

Corollary 2 *The Match-and-Split(G,H) algorithm's running time is $O(n_G n_H + (n_G + n_H) m_G m_H)$ when the local matching criterion is similar length-1 paths.*

Our proof of the two running time bounds rests on some analysis of the algorithm's recursion structure, which we present now as a lemma before the proof. To state the lemma, we need to clarify some terms related to the recursion call tree of the *Match-and-Split*(G,H) algorithm. In this tree, a leaf is any algorithm call that terminates recursion (including any that outputs a maximal solution), an internal node is any call that invokes other calls (its children) recursively, and a level is a set of calls at the same recursion depth from the initial call on the input graphs (Figure 3.3 provides some illustrations). Note how the number of leaves is an upper bound on the number of output solutions.

Lemma 3 *The number of leaves in the recursion call tree of the Match-and-Split(G,H) algorithm (and hence the number of output solutions) is at most $n_G n_H$. The number of internal nodes in this tree is also at most $n_G n_H$, and the number of levels is at most $n_G + n_H$.*

Proof (of Lemma 3): First, we prove the quadratic bound on the number of leaves by arguing any node pair in G, H is present in at most one leaf of the call tree (and hence in at most one output solution too). As the G_i's and the H_j's are node-disjoint, a call on G, H *partitions* the node pairs in G, H among its children calls on G_i, H_j for all $i = 0$ to $c, j = 0$ to d (where we add dummy leaf children involving $G_0 \doteq G - G'$ or $H_0 \doteq H - H'$ for analysis, when these graphs are non-empty). If a node pair is in two leaves, then the least common ancestor of the leaves should have sent this same node pair

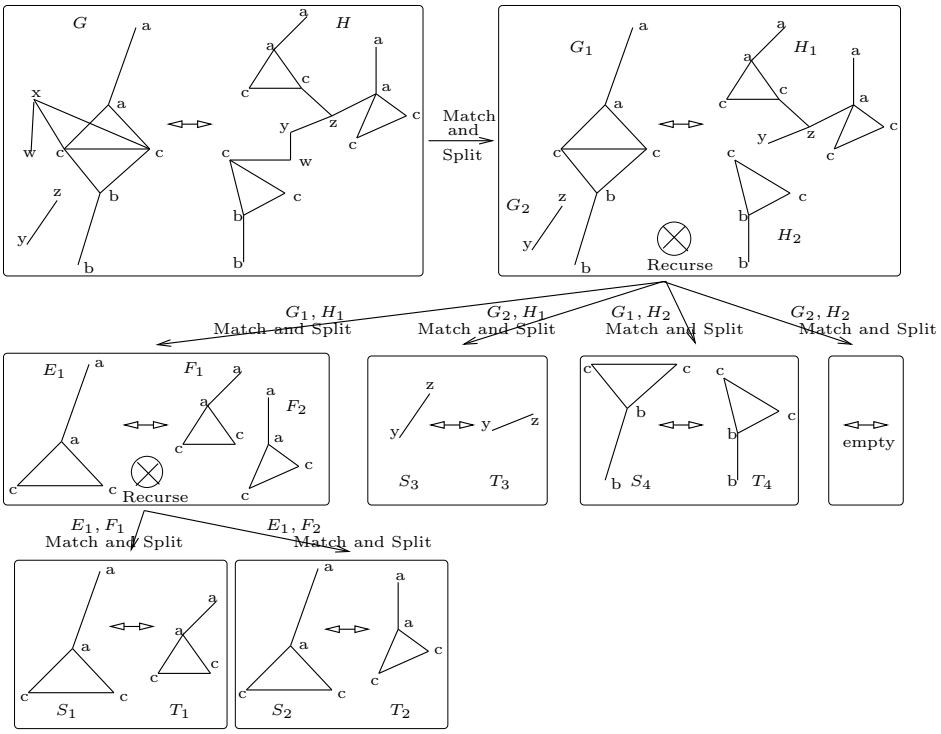

Figure 3.3. *Match-and-Split* algorithm's recursion call tree on sample input graphs G, H, using similar length-1 paths as local matching criterion. Only node labels are shown to reduce clutter, and $sim(u, v)$ is true whenever labels of nodes u, v shown beside the nodes are the same. The subgraph pairs S_i, T_i are the solutions found.

along two of its children, a contradiction! Next, observe that the number of internal nodes is at most the number of leaves as each internal node has at least two children (including dummy leaves). Finally, the number of levels is at most $n_G + n_H$ as each child (call on G_i, H_j) of any internal node (call on G, H) satisfies $n_{G_i} + n_{H_j} \leq n_G + n_H - 1$. If it were not true, then $n_{G_i} + n_{H_j} = n_G + n_H$, thereby making G, H a maximal solution and this internal node on G, H a leaf instead! □

Proof (of Theorem 1): To derive the bound for a general $f(\cdot, \cdot)$, note simply that the number of recursion calls made by the algorithm is at most $2\, n_G n_H$ from Lemma 3 and the process time at each call is bounded by a loose $f(G, H)$ (as we may assume $f(\cdot, \cdot)$ is a non-decreasing function).

To derive the special bound, we also use the bound on the number of levels from Lemma 3. The two terms in the special bound are: $O(n_G n_H\, f_0)$ for the f_0 time spent at each of

the $O(n_G n_H)$ recursion calls made, and $O((n_G + n_H)(f(G,H) - f_0))$ for the $(f(G,H) - f_0)$ remaining process time at each of the $O(n_G + n_H)$ *levels* (assuming the special condition). We derive the latter term to complete the proof.

Consider the class of functions $f(\cdot,\cdot)$ that satisfy the special condition $\sum_{i=1}^{c}\sum_{j=1}^{d}(f(G_i,H_j) - f_0) \leq (f(G,H) - f_0)$ for any G, H. For such $f(\cdot,\cdot)$, we now bound the process time (of the match and split steps, excluding the constant f_0 terms) at each level of the recursion call tree. The condition above simply says that the process time at *all* children of an internal node in the call tree is at most the process time at the internal node itself. We partition the calls (nodes) in a level l into sibling groups, and bound the time of each sibling group by that of their single parent in level $l-1$. So the time at all nodes in a level is at most that in the preceding level. Cascading these relations, the process time at each level is at most that at the root level call on G, H, which is $(f(G,H) - f_0)$. □

Generality of the algorithm

Our algorithm framework admits quite general schemes to search for similar subgraph pairs and score them, and is hence attractive in a biological setting. The searching scheme is flexible as our problem variant and algorithm works for different connectivity and local matching criteria. The scoring scheme is flexible as it is decoupled from the searching scheme. Besides, the number of maximal solutions is only quadratic, so it is not expensive to compute a sophisticated, biologically-inspired score for every solution.

We discuss the flexibility of the local matching and connectivity criteria in more detail. We already saw different monotone local matching criteria. We could also combine such monotone criteria to get a new monotone criterion. For example, declaring two nodes as locally matching if they are so with respect to similar length-p paths *or* s-similar neighborhoods gives a less stringent criterion. Many connectivity criteria are possible too. We could replace connectedness with biconnectedness (Baase, 1991) for instance, by simply changing the algorithm's split step and still obtain a provably efficient algorithm. The number of output solutions is then at most $m_G m_H$ and the running time $O(m_G m_H f(G,H))$, the

proofs of which (omitted) are a simple extension of the connectedness proofs and use the property that the biconnected components of a graph are edge-disjoint.

3.1.3 Overall Method

Our method of detecting conserved modules between two protein networks involves a searching scheme to find similar subgraph pairs (*candidate conserved modules* or *candidates*) using our graph-matching algorithm above, and a scoring scheme to rank these candidates using statistics of similar paths between a subgraph pair. We now describe these schemes and their place in the overall *Match-and-Split* method.

Searching scheme (via graph-matching)

To adapt the generic graph-matching algorithm *Match-and-Split* to the specific task of comparing protein networks, we make default choices for certain graph-matching parameters and incorporate a clustering heuristic to handle solutions that are large. In this section, our algorithm's maximal solutions are referred to simply as *solutions*.

Choosing parameters. Our default parameter choices result in a lenient graph-matching criterion, for we would like to detect as many functional modules as possible from noisy protein networks of divergent organisms (e.g. yeast and human). The default choices we made on exploring a limited parameter space follow.

Connectedness defines our connectivity criterion. We choose it over biconnectedness as some functional modules (e.g. linear signaling pathways) are not biconnected, and even those over highly interacting proteins (e.g. protein complexes) may not appear biconnected due to incomplete interaction data.

Similar length-p paths (for $p = 1, 2$) defines our local matching criterion. We choose it over s-similar neighborhoods (for $s = 1, 2$) again on the basis of sensitivity. For $p, s = 1$, both options are equivalent as they yield the same $lm(S, T)$ node-set (defined in Section 3.1.2).

However for $p, s = 2$, this node-set from similar paths is a superset of the one from similar neighborhoods, so similar paths is a more lenient criterion.

Sequence similarity defines our node similarity function as in previous network comparison methods. For fair evaluation, we in fact choose the same criteria used in two previous methods: (A) $sim(u, v)$ is true whenever the BLAST E-value of proteins u, v is at most 10^{-7} and each protein is among the 10 best BLAST matches of the other (Sharan *et al.*, 2005), and (B) $sim(u, v)$ is true whenever the BLAST E-value of u, v is less than that of 60% of ortholog pairs in some ortholog database (see (Koyuturk *et al.*, 2006) for details).

Incorporating clustering heuristic. Increased sensitivity from the lenient graph-matching criterion above comes at a cost. Sometimes, a solution is over a large number of proteins and hence less specific. For instance, a solution from a preliminary comparison of yeast and human networks covers more than 500 yeast and human proteins! To split such large solutions, we incorporate a betweenness clustering heuristic in our *Match-and-Split* algorithm. This clustering splits a graph into highly-connected, smaller clusters based on iterative computations of an edge betweenness centrality measure (Girvan and Newman, 2002) (see Supplemental Text A for details). One could also cluster a graph using other methods such as the popular spectral clustering methods (Weiss, 1999).

We incorporate the clustering by replacing our algorithm's '[base-case]' statement with the code block below. As before n_G refers to the number of nodes in G, and we may assume $n_G \geq n_H$ without loss of generality. The parameter $n_{\max}$ (say 25) indicates when a solution is large.

[base case] code block:
if ($n_G \leq n_{\max}$ and $n_H \leq n_{\max}$)

Output the maximal solution G, H.
else [large solution]

Split G into clusters $G_1, \ldots, G_e$ using betweenness clustering.

for $i = 1$ to e

Match-and-Split(G_i, H).

A betweenness clustering of G takes $O(n_G m_G^2)$ running time (Girvan and Newman, 2002). In practice, incorporating the clustering heuristic is not expensive as our experiments result in very few large solutions (mostly one), each covering just a few hundred nodes.

Scoring scheme (via similar paths)

We score a candidate conserved module based on the number of similar length-p paths, and express the statistical significance of the score as a P-value. We use the P-values both to rank the candidates from the searching scheme and to retain only those with P-values at 10% significance level after multiple testing. Our scoring scheme is flexible, as seen in Section 3.1.2, in permitting complicated scoring measures including measures used in previous network comparison methods. Still we use a simple scoring measure and the promising results we obtain show the strength of our searching scheme based on the graph-matching algorithm.

The score of a candidate conserved module $S \subseteq G, T \subseteq H$, where G, H are the input protein networks, is simply the number of pairs of similar length-p paths between them (defined in Section 3.1.2). We evaluate the P-value of this score using a null model that randomizes the edges and node similarity function of G, H to exclude the mechanism of interest viz., conservation of protein modules. To provide a stringent control, the randomization loosely preserves the degree sequence and node similarity distribution as in previous methods. The simplicity of our scoring measure and null model allows us to develop an analytical bound on the P-value (see Supplemental Text A). This bound can also incorporate reliabilities of noisy protein interactions.

Implementation pipeline and dataset

The overall Match-and-Split method proceeds in a pipeline to detect candidate conserved modules between two protein networks. First our searching scheme uses the graph-matching algorithm to produce candidates, then a size filter retains only medium-sized

candidates, and finally our scoring scheme ranks the candidates by their P-values and retains those at 10% significance level (after multiple testing). A fast implementation of the method is publicly available (see Supplemental Text A for website reference), and it takes only a few minutes (at most 4) on a 3.4 GHz Pentium Linux machine to compare two studied networks.

The size filter, similar to one in a previous method (Sharan *et al.*, 2005), retains any candidate subgraph pair S, T whose number of nodes n_S, n_T satisfy $n_{\min} \leq n_S, n_T \leq n_{\max}$ ($n_{\min} = 3, n_{\max} = 25$ in our experiments, and $n_{\max}$ is same as in the searching scheme's clustering heuristic). We focus on such medium-sized candidates for the following reasons. A large module as a whole is likely to correspond to a less specific function and worse causes artifactual increase in sensitivity in our evaluation studies. A small module, over say two proteins, is likely to result from a spurious match occurring simply by chance.

The protein networks for model organisms *Saccharomyces cerevisiae*, *Drosophila melanogaster* and *Caenorhabditis elegans* (referred hereafter as yeast, fly and worm respectively) are experimentally-derived (e.g. two-hybrid, immunoprecipitation) interactions collected in the DIP database (Salwinski *et al.*, 2004). The version of the networks used and the interaction reliabilities based on a logistic regression model are taken from a previous study (Sharan *et al.*, 2005). Our human protein network is from the HPRD database (Peri *et al.*, 2003) (binary or direct interactions; Sep 2005 version), and we assume unit reliability for these interactions as they are literature-based.

3.1.4 Evaluation Measures

We describe some measures that we would need in the Results section to evaluate and interpret the conserved modules from pairwise network comparisons. The notation *yeast-human comparison* denotes the comparison between yeast and human protein networks, and *yeast-human modules* denotes the resulting conserved modules. Similar notations apply for other two-species comparisons too.

Performance (sensitivity and specificity)

The performance of a pairwise network comparison method depends on the quality of candidate conserved modules it outputs, which we could measure by how well the candidate set aligns with a reference set of conserved modules. But such reference sets, even if available, are not comprehensive or applicable for our purpose (e.g. STKE (`http://stke.sciencemag.org/cm/`, Jun 2005) contains only a couple of fly-worm conserved signaling pathways and no protein complexes; Biocarta (`http://www.biocarta.com/genes/allPathways.asp`, Jan 2006) has many human-mouse conserved pathways but available mouse interaction data is sparse).

A viable alternative is to use known functional modules in a *single species* as reference to evaluate part of a candidate set. Our reference modules are the literature-based yeast protein complexes present in MIPS (Mewes *et al.*, 2004) (May 2006 version), at level at most 3 in its complex category hierarchy. We consider only complexes with 3 to 25 proteins due to our study's focus on medium-sized modules (see Section 3.1.3 for reasons). To test a method, we compare the yeast network against the human, fly or worm network (dataset details in Section 3.1.3), collect the yeast subgraphs S from each output candidate S, T, and measure the overlap of these single-species candidate modules with the reference modules.

Our main measures are module-level sensitivity and specificity, which are the fraction of reference modules covered by some candidate module and the fraction of candidate modules covered by some reference module respectively. A module S of proteins from a single species (yeast) is covered by another module S' if $|S \cap S'|/|S| \geq 50\%$, a stringent criterion given the noisy interaction data.

We also present measures at the interaction and protein levels to assess the quality of different components of candidate modules (similar to measures at the gene, exon and nucleotide levels in gene structure prediction methods (Burset and Guigo, 1996)). Define the protein interactions *spanned* by a set of modules as the union of interactions present in each of these modules. If set A denotes the interactions spanned by all candidate modules

and B that by all reference modules, then interaction-level sensitivity is $|A \cap B|/|B|$ and specificity is $|A \cap B|/|A|$. Protein-level measures are defined similarly.

GO analysis measures

We use the GO resource (The Gene Ontology Consortium, 2000) for functional annotations of genes. Specifically, all our functional descriptions are based on known GO Biological Process annotations (Aug 2006 version) of proteins to terms at level at least 4 in the GO hierarchy. Given these annotations, we next define many GO-related concepts that we would need later.

The *best GO term* of a protein module is the term the module's proteins are enriched for with the least hypergeometric P-value (computed by GO::TermFinder (Boyle *et al.*, 2004) at 10% significance level with Bonferroni correction). Given two GO terms, the *match* between them is the overlap $\min(|A \cap B|/|A|, |A \cap B|/|B|)$ between the set A, B of terms they imply, where a GO term implies itself and its ancestors in the GO ontology. This match is based on a well-justified measure (Kiritchenko *et al.*, 2005).

We call a protein module *functionally homogeneous* if at least 50% of its proteins are annotated with the best GO term the whole module is enriched for. We call a candidate conserved module S, T (e.g. yeast-human candidate) as *functionally similar* with respect to GO if S and T are each functionally homogeneous and the match between their best GO terms is at least 75%.

When validating a predicted GO term of a protein, we use a well-justified criterion as for match between terms. This criterion requires the respective sets A, B of terms implied by the predicted term and some known term annotated to the protein to have a high overlap $|A \cap B|/|A| \geq 75\%$.

3.2 Results

3.2.1 Performance against Previous Methods

We test our Match-and-Split method against two previous methods, NetworkBLAST and MaWISh. We try two Match-and-Split versions, $p=1$ and $p=2$, which specify the local matching criterion of similar length-p paths. We use the default versions of NetworkBLAST and MaWISh that detect conserved protein complexes or subnetworks (see Supplemental Text A for software references). We don't test the Bayesian alignment (Berg and Lassig, 2006) and Græmlin (Flannick *et al.*, 2006) methods as they are from very recent studies with results focusing on coexpression or prokaryotic networks, whereas the current study's focus is eukaryotic protein networks. Fundamentally though, all these methods work for the networks of any species.

We evaluate a method by measuring how well the yeast modules it outputs (on pairwise protein network comparisons of yeast and other species) align with a reference set of yeast protein complexes in MIPS, as explained in Section 3.1.4. For fair evaluation, we attempt to use common input, default parameter values, and common output processing for all methods. For instance, the input networks *and* node similarity function $sim(\cdot,\cdot)$ are the same across methods. The $3, 25$ size filter thresholds are the same too, so we evaluate all methods only on the *medium-sized candidates* they output (see Section 3.1.3).

Different $sim(\cdot,\cdot)$ criteria could yield quite varying results, and we try two criteria used in previous methods as discussed in Section 3.1.3. The main text presents criterion A results, and the Supplemental Text A presents some criterion B results to show few changes in the relative performance of the methods between the two criteria (mainly in yeast-fly comparison where module-level results of NetworkBLAST are better than other methods with criterion B and not A).

First, we discuss module-level results of Match-and-Split ($p=1$) relative to the other two methods. In yeast-human comparison (Table 3.1), Match-and-Split and MaWISh have comparable performance with Match-and-Split being somewhat better, and NetworkBLAST

Method	# output modules (interactions, proteins)	Module		Interaction		Protein	
		Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
Match-and-Split							
(p=1)	80 (667, 421)	25.0	47.5	20	35	25	48
(p=2)	72 (664, 411)	28.0	41.7	20	34	25	47
NetworkBLAST	421 (1311, 606)	40.9	18.5	34	30	37	48
MaWISh	151 (508, 389)	20.5	43.7	15	33	23	46

Table 3.1. Evaluation of output candidates from yeast-human network comparison using sensitivity (sens.) and specificity (spec.) measures (expressed as rounded percentages) at the module, interaction and protein levels. The second column shows the number of yeast modules (candidates) output, and the number of interactions and proteins spanned by these yeast modules. The reference set comprises 132 medium-sized (size 3 to 25) yeast complexes in MIPS that span 1144 interactions and 791 proteins. All relevant definitions are in Section 3.1.4.

Method	# output modules (interactions, proteins)	Module		Interaction		Protein	
		Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
Match-and-Split							
(p=1)	27 (155, 123)	6.8	51.9	5	35	8	49
(p=2)	25 (131, 110)	7.6	48.0	4	37	7	53
NetworkBLAST	77 (354, 206)	8.3	39.0	9	28	12	46
MaWISh	26 (81, 87)	5.3	50.0	3	40	6	52

Table 3.2. Evaluation of output candidates from yeast-fly network comparison, using the same format as Table 3.1.

has better sensitivity than other methods at the cost of very low specificity and an output of too many candidates. A similar though less pronounced trend appears in the yeast-fly case (Table 3.2). In the yeast-worm case (Table 3.3), NetworkBLAST has much better specificity than other methods at comparable sensitivity.

Moving from relative to absolute performance, the values of module-level sensitivity and specificity are low (for example, a mere 25.0% and 47.5% respectively by Match-and-Split (p=1), a competitive method in yeast-human case). Low sensitivity could be due to factors like noisy interaction data, poor conservation of complexes across compared organisms,

Method	# output modules (interactions, proteins)	Module		Interaction		Protein	
		Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
Match-and-Split							
(p=1)	17 (116, 76)	3.8	58.8	5	53	5	54
(p=2)	12 (126, 67)	4.6	50.0	6	58	5	57
NetworkBLAST	27 (182, 74)	3.8	81.5	8	51	6	64
MaWISh	24 (78, 61)	2.3	70.8	4	54	4	54

Table 3.3. Evaluation of output candidates from yeast-worm network comparison, using the same format as Table 3.1.

or differing computational and biological definitions of a functional module or complex. Finding which of these factors is key is a subject of future work. Low specificity is probably due to incompleteness of the reference set, and it does not indicate the output candidates are spurious (a claim supported by further analysis in Section 3.2.4).

3.2.2 Single-species vs. Pairwise Network Analysis

Pairwise network comparison methods use cross-species conservation in an attempt to improve over single-species methods in detection of functional modules. But previous network comparison studies have not evaluated their methods against single-species ones. Here we undertake this evaluation by testing a popular single-species method MCODE (Bader and Hogue, 2003) on the yeast network under the same measures and size range (3 to 25 proteins) as before. It is beyond the scope of this study to test all single-species methods.

The performance of the pairwise methods relative to the single-species MCODE is varied. Under proper homolog and species selection, Match-and-Split performs better than MCODE in detecting reference yeast complexes. Comparing Table 3.1 on yeast-human comparison with Table 3.4, Match-and-Split (p=1) is much more sensitive than MCODE at the same specificity, and MaWISh performs similar to MCODE. The choice of homologs is important as changing the $sim(\cdot,\cdot)$ criterion from A here to B in Table A.1 reduces the benefit of pairwise methods over MCODE. Species selection is also crucial as the pairwise methods have worse sensitivity than MCODE in the yeast-fly (Table 3.2) and yeast-worm (Table 3.3)

Method	# output modules (interactions, proteins)	Module		Interaction		Protein	
		Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
MCODE	53 (614, 323)	16.7	47.2	19	36	20	48

Table 3.4. Evaluation of output clusters from yeast network analysis by MCODE, a single-species clustering method, using the same format as Table 3.1. We focus on medium-sized (size 3 to 25) clusters as in other evaluations.

Method	# output modules (interactions, proteins)	Module		Interaction		Protein	
		Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
Match-and-Split	81 (634, 410)	24.2	48.2	20	35	25	49
Split-only	569 (3597, 2776)	50.0	12.3	58	18	76	22

Table 3.5. Evaluation of Match-and-Split ($p = 1$) on pairwise yeast-human network comparison against Split-only on single-species yeast network clustering, again using the same format as Table 3.1. Similar results from Match-and-Split ($p = 2$) is omitted. As betweenness clustering of large graphs is compute-intensive, Split-only uses a quicker version of it (see Supplemental text; Split-only still requires orders of magnitude more time than Match-and-Split). For fairness, the Match-and-Split version here uses the quicker clustering inside its searching scheme.

cases. Low sensitivity in the yeast-worm case could be due to the sparse worm interaction data, but the reason in the yeast-fly case is unclear as the fly network is interaction-rich.

Our Match-and-Split searching scheme includes a betweenness clustering heuristic. The heuristic is by itself a single-species method (denoted Split-only) for it can cluster the full yeast network into highly connected modules. Table 3.5 compares Match-and-Split and Split-only to show the boost in specificity from adding pairwise match criterion to plain clustering. Split-only detects more reference yeast complexes but at the cost of outputting too many candidate modules at very low specificity. Adding pairwise match criterion also drastically reduces running time by restricting the size of graphs that need to be betweenness clustered.

Rank	P-value (score)	Size	Best GO term of module (% annotated proteins)		
			Yeast	Human	Terms' match
1	3.33e-13 (8)	5, 3	purine ribonucleoside salvage (100%)	nucleoside metabolism (100%)	53%
2	1.05e-12 (3)	3, 4	protein import into mitochondrial matrix (100%)	protein targeting to mitochondrion (75%)	89%
3	1.38e-12 (4)	3, 3	postreplication repair (100%)	DNA repair (100%)	94%
4	1.40e-12 (6)	4, 3	ER to Golgi vesicle-mediated transport (100%)	ER to Golgi vesicle-mediated transport (100%)	100%
5	1.89e-12 (2)	3, 3	processing of 20S pre-rRNA (100%)	rRNA processing (100%)	95%

Table 3.6. Five top-ranked candidates from Match-and-Split ($p=1$) yeast-human comparison. The size of a candidate (third column) is the number of its yeast, human proteins. The '% annotated proteins' is the fraction of proteins in a module annotated with the module's best GO term, and the match shown is between the best GO terms of yeast module S and human module T in a candidate S, T (see Section 3.1.4 for definitions).

3.2.3 Select Conserved Modules

The results above evaluate the candidates from our method on a global scale. Here we discuss the biology of a select few candidates. We start with a flavour of some top-ranked candidates from the Match-and-Split ($p=1$) yeast-human comparison in Tables 3.6 and 3.7. These tables contain functional descriptions based on some known GO Biological Process annotations as described in Section 3.1.4. The format of these tables is inspired from a previous study (Koyuturk *et al.*, 2005).

Consider the candidate ranked 2 in Table 3.6 and shown in Figure 3.4. From literature-based descriptions in SGD (Hong *et al.*, 2006) and UniProt (Apweiler *et al.*, 2004), the yeast and human proteins of this candidate are each components of the TIM23 complex, a mitochondrial inner membrane translocase. This complex mediates translocation of preproteins across the mitochondrial inner membrane. Typical preproteins are nuclear-encoded, synthesized in the cytosol and contain a targeting sequence (presequence or transit peptide) to direct transport.

Rank	P-value (score)	Size	Best GO term of module (% annotated proteins)		
			Yeast	Human	Terms' match
50	7.76e-10 (40)	12, 10	ubiquitin-dependent protein catabolism (100%)	ubiquitin-dependent protein catabolism (100%)	100%
72	1.02e-08 (16)	12, 12	DNA-dependent DNA replication (75%)	DNA metabolism (91.7%)	83%
74	1.48e-08 (95)	16, 17	protein amino acid phosphorylation (81.2%)	phosphorus metabolism (88.2%)	35%
77	6.40e-08 (18)	15, 13	transcription initiation (86.7%)	transcription initiation (69.2%)	100%
78	1.28e-07 (79)	20, 23	actin filament organization (65%)	Rho protein signal transduction (43.5%)	11%

Table 3.7. Five top-ranked candidates with at least 10 yeast and 10 human proteins from Match-and-Split ($p = 1$) yeast-human comparison, presented in the same format as Table 3.6.

We now elaborate on the candidate ranked 72 in Table 3.7 and shown in Figure 3.4. The candidate may seem too heterogeneous to be conserved but it actually contains many homologous complexes as inferred from literature-based comments at SGD and UniProt. This example illustrates how a lenient matching criterion over noisy interactions can detect known complexes. The origin recognition complex (of the ORC proteins) with counterparts in yeast and human binds replication origins, and plays a role in DNA replication and transcriptional silencing. The RAD1, RAD10 and RAD14 proteins are subunits of the Nucleotide Excision Repair Factor 1 (NEF1) in yeast and homologous to the ERCC4, ERCC1 and XPA respectively in human. The RFA1, RFA2 in yeast (homologs RPA1, RPA2 in human) are subunits of heterotrimeric Replication Factor A (RF-A), a single-stranded DNA-binding protein involved in DNA replication, repair and recombination.

3.2.4 Annotation Transfer from Yeast to Human

The candidates output by pairwise network comparison methods, like the ones sampled in last section, enable transfer of functional annotations between organisms. The idea is to annotate a protein module in one organism with a function that a similar module

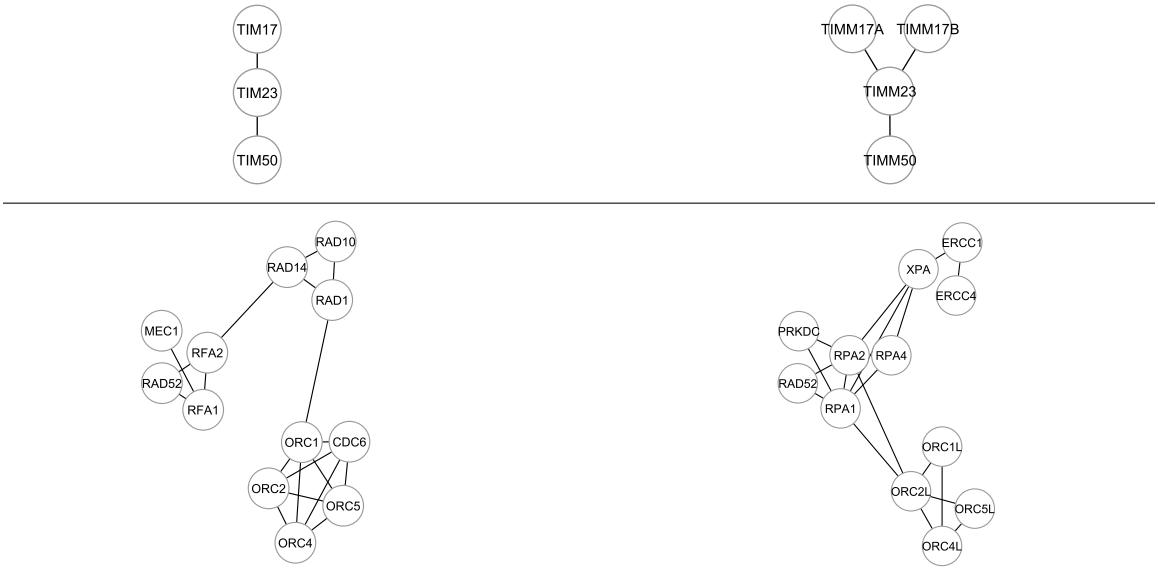

Figure 3.4. Select candidates from Match-and-Split ($p=1$) yeast-human comparison. Each candidate is a conserved module of yeast (left) and human (right) proteins. Two proteins similar by the $sim(\cdot,\cdot)$ function are roughly aligned horizontally.

in another organism is known to be enriched for. Our focus here is annotation transfer from yeast to human based on candidate conserved modules (i.e., yeast-human candidates) output by Match-and-Split ($p=1$), using certain known GO Biological Process annotations described in Section 3.1.4. The Match-and-Split results in this section are competitive with the corresponding results of other tested methods (shown in Tables A.2, A.3).

We first present results on functional homogeneity and similarity of yeast-human candidates (as defined with respect to GO in Section 3.1.4). Collect the yeast module S of every yeast-human candidate S, T. Then all (100%) of these yeast modules are homogeneous for some function, which could then be transferred. This fraction is 83.8% for the human modules collected similarly. The fraction of yeast-human candidates functionally similar with respect to GO is a reasonable 42.5%, which further supports annotation transfer.

The actual transfer on each yeast-human candidate S, T involves assigning all human proteins in T to the best GO term the yeast module S is enriched for. If this procedure predicts more than one GO term for a human protein, we retain the term with the least hypergeometric P-value (described in Section 3.1.4). We predict GO Biological Process terms for a total of 462 human proteins (predictions available online; see Supplemental

text). This annotation transfer is reliable as a reasonable 295 of these predictions are valid by a stringent, well-justified criterion in Section 3.1.4. This transfer covers only 462 proteins though, a small fraction of the 1882 proteins in the human network sequence-similar to some protein in the yeast network (by the $sim(\cdot, \cdot)$ function).

3.3 Discussion

In the context of comparing two protein networks, this work shows it is possible to design a provably good search algorithm that also translates to promising performance in practice. The algorithmic guarantees of our Match-and-Split method distinguishes it from previous methods based on greedy heuristics. Formal guarantees are important as they lend credibility to the conserved modules found by a bounded-time search procedure.

Our method Match-and-Split performs competitively in tests against two previous pairwise network comparison methods. For instance, Match-and-Split performs comparably to or somewhat better than the MaWISh method. In tests against a single-species method MCODE, Match-and-Split performs better in yeast-human comparison. This single-species test, which was not done in previous pairwise studies, also reveals comparisons (yeast-fly and yeast-worm) where the pairwise methods are poorer than MCODE.

The above evaluations, especially the single-species one, lead to an immediate future question. Are the poor results of pairwise methods in some comparisons mainly due to incomplete interaction data or something intrinsic to the choice of the species pair and homologs between them? The answer could inform the conditions when pairwise methods exploit cross-species conservation to improve single-species detection of functional modules.

The conserved modules our method detects and their functional descriptions are the findings of most practical interest to biologists. Also of interest are the reasonably accurate functional predictions resulting from the transfer of GO annotations between conserved modules. We make these findings publicly available (at a website mentioned in the Supplemental text), along with a fast Match-and-Split implementation to facilitate new network comparisons.

Our graph-matching algorithm is flexible in allowing diverse local matching and connectivity criteria. A biologist could for instance design a stringent matching criterion to detect similar instances of a functional module in a single-species network (duplicated modules). We could even compare other types of networks like metabolic networks within the same algorithmic framework. A challenge then is the judicious design of a matching criterion for the biological comparison of interest.

Our algorithm guarantees are limited to monotone local matching criteria. Many useful criteria, such as one that declares two nodes locally matching if they are similar and at least half of their neighbors are similar, are non-monotone. A future direction is to explore tractable search formulations for non-monotone criteria. Another limitation with the current study, but not with our framework, is the use of a simple scoring scheme. The simple scores yield reasonably good results, however network conservation scores that correlate better with the biological significance of a conserved module is a subject of future work.

Chapter 4

Probabilistic Estimation of the Phylogeny of a Pathway

We design a probabilistic model for pathway evolution under the hypothesis that the pathway is evolving as a heritable unit across closely related species, and apply it to estimate the phylogenetic tree topology along which the pathway evolved from its unknown ancestral forms to present-day forms in the different species. An overview of our probabilistic approach and its contributions relative to other discrete similarity based approaches was presented in Chapter 2. This chapter discusses in depth our evolutionary model, a distance matrix tree estimation method based on the model, and its application to estimate the phylogeny of several microbial pathways. The design of our model involves extending a previously published 2-character co-evolution model (Barker and Pagel, 2005; Pollock *et al.*, 1999) to handle $k > 2$ characters (genes), and then achieving a tractable number of parameters via a Markov network (Jordan, 1999) representation of the pathway data, and an assumption about uniform evolutionary rate for genes. Our results include interesting hypotheses about the evolution of certain bacterial and archaeal pathways, which are derived from interpreting the estimated pathway trees using sequence-based species trees and data on cellular phenotypes.

4.1 Evolutionary Model

We develop our evolutionary model in this section, and show how to use it for phylogenetic estimation in later sections. Our model of pathway evolution captures how the genes (nodes) in a pathway are gained or lost over time in some correlated fashion, by using cross-species gene content data and canonical interaction (edge) data.

An interaction between two pathway genes is termed *canonical* if it is observed in some model organism where the pathway is studied in detail. For example, researchers often take a canonical view of the interactions in a metabolic pathway based on detailed studies of the pathway in *E. coli*. That is, they assume that an interaction between two *E. coli* genes (enzymes) also exists between the orthologous counterparts of these genes (if both are present) in other closely related species. In metabolic pathways, an interaction between two genes denotes that their enzyme products catalyze contiguous chemical reactions. In general, the interaction between two pathway genes could also denote a physical interaction between their protein products, or a direct functional association such as that between a transcription factor and its target gene.

4.1.1 Pathway Representation

We seek a biologically plausible model for the evolution of a pathway as a single coherent unit. We also prefer a model with few or constant number of parameters for reasons mentioned in Section 2.2. To obtain such a tractable model, we could reduce the representation of the pathway to include only the presence or absence of its nodes and edges in different species. But note that for a typical pathway, only node data is available for several species (e.g. gene content of microbial species with sequenced genomes), and edge data is available for just a few model organisms where the pathway interactions are studied in detail.

Taking all this into account, our model represents a pathway by the presence or absence of genes in different species, and uses canonical interactions between pathway genes observed in model organisms to specify fine-grained dependency among gene gains and losses. So a pathway of k genes can be in one of the 2^k states indicating the presence or absence of

(orthologs of the) genes in a species, and its state evolves under some static dependence constraints imposed by the canonical pathway edges. In other words, our pathway evolution model is really a pathway gene content co-evolution model with static dependence structure defined by the pathway edges (note that the term canonical is sometimes dropped when the meaning is clear).

4.1.2 Co-evolution Model of k Characters

In this section, we develop a general model for the correlated evolution of k binary characters along a phylogenetic tree. To obtain the specific pathway evolution model outlined above, we let the characters indicate the presence or absence of the k pathway genes, and parameterize the model using a tractable number of free parameters that reflect the biology of the evolving pathway (as detailed in the next section).

Model parameters and intuition. Our general model assumes that the static dependence constraints on the evolution of the k characters are provided as a joint distribution π over the characters (defined implicitly by a few parameters), and the character evolution rates are provided as a parameter set $\lambda = \{\lambda_i\}_{i=1}^{k}$. To clarify, let the state space of size 2^k of the k binary characters be denoted as $\{u : u \in \{0,1\}^k\}$. Then in our model, the joint distribution $\pi = \{\pi_u\}$ over the characters dictates the correlated fashion in which these characters evolve over time, and the parameter λ_i dictates the rate at which character i evolves. The ensuing text describes in detail how the model uses these parameters to define dependent evolution of the characters.

Substitution model along a phylogeny. Evolution of characters along a tree topology with branch lengths can be modeled along the lines of standard phylogenetic models of nucleotide sequences (Felsenstein, 2003). The question, then, is how to design a substitution model for the co-evolution of k characters. A substitution model specifies how characters change (substitute) from one state into another along a branch of the tree (Figure 4.1). For example, when the characters are nucleotides, there are four substitution states: A,

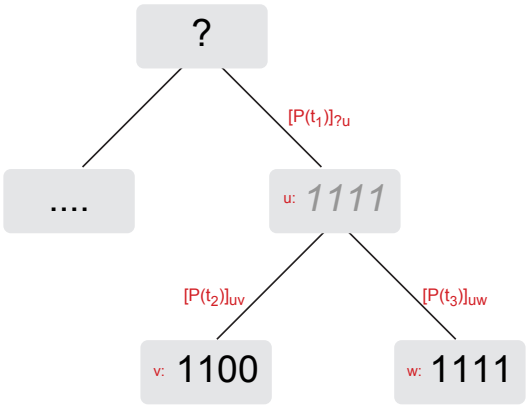

Figure 4.1. Binary characters co-evolving along a given tree. The transition probabilities along a branch of length t are denoted by the matrix $P(t)$. The leaves of the tree correspond to the state of characters observed in current-day species. The internal nodes correspond to unobserved ancestral states. The example ancestral state is shown only for illustration. The last two characters might be co-evolving as they are both lost during a short period of time.

C, G, and T. In our case, the k characters can be in one of the 2^k possible states, and the substitution model is fully specified by a continuous time Markov chain over these 2^k states. The transition probabilities of the Markov chain along a branch of length t is given by $P(t) = \exp(Rt)$ (Figure 4.1), where R is the $2^k \times 2^k$ rate matrix associated with the chain. We define R next using the (π, λ) parameters to complete our model description.

The rate matrix R. Our rate matrix R, which underlies the evolutionary model, is an extension of the 2-character correlated evolution rate matrix (Barker and Pagel, 2005; Pollock *et al.*, 1999) to the general k-character case. Being used in a model of dependent evolution of characters, this rate matrix also shares some aspects of rate matrices of nucleotide substitution models that allow dependence between sites, such as the codon model of Goldman and Yang (Goldman and Yang, 1994), and the recent site dependence models of Jensen and Pedersen (Pedersen and Jensen, 2001) and Robinson *et al.* (Robinson *et al.*, 2003). As with dependent-sites models, our sparse rate matrix assumes a zero instantaneous rate for simultaneous changes in more than one character; note however that there is non-zero probability for multiple successive changes along a branch of the tree. The design

of biologically realistic rate matrices that capture dependence between characters or sites remains an active open area of research.

We are ready to define our $2^k \times 2^k$ rate matrix R. Recall the state space and model parameter notations from above, and let $\mathrm{h}(u, v)$ denote the number of characters in which states u, v differ (i.e., their Hamming distance). Then

$$\begin{aligned} R(u,v) &= 0 \qquad \text{if } h(u,v) \geq 2, \\ R(u,v) &= \lambda_i \frac{\pi_v}{\pi_u + \pi_v} \quad \text{if } h(u,v) = 1; u, v \text{ differ in character } i. \end{aligned}$$

For example, $R(1111, 1100) = 0$, and

$$R(1111, 1101) = \lambda_3 \frac{\pi_{1101}}{\pi_{11*1}} = \lambda_3 \frac{\pi_{1101}}{\pi_{1111} + \pi_{1101}} \ .$$

As a model of co-evolution, this rate matrix has some desirable properties. It is time-reversible with π as its stationary distribution, and captures the dependence between evolution of characters as specified in π. That is, the rate $R(1111, 1101)$ is proportional to the conditional probability under π that the third character is absent when all other characters are present. In general, this can be different from $R(0010, 0000)$, permitting the rate of change of the third character to depend on the background state of the other characters. The conditional probability formulation also lets the Markov chain factorize according to the dependence structure in π. For example, if the third character were independent of other characters under π, this forces all rates of loss of the third character to the same value (e.g. $R(1111, 1101)$ becomes equal to $R(0010, 0000)$). Thus the overall Markov chain factorizes into two independently evolving Markov chains: one for the third character and another for the other three.

4.1.3 Co-evolution Model of Pathway Gene Content

To adapt the general k-character co-evolution model to our case of pathway evolution outlined in Section 4.1.1, we let the characters correspond to the presence or absence of k genes in the pathway, and set the parameters (π, λ) suitably to reflect the correlated gains

or losses of interacting genes. We parameterize π by formulating a Markov network (Jordan, 1999) from the canonical pathway edges, and parameterize λ by assuming a uniform rate of gain or loss for the genes.

A contribution of this parameterization is the constant number of free parameters it requires. A tractable parameter set is necessary for the purpose of phylogenetic estimation, because a full parameterization requires 2^k parameters for π and k parameters for λ (in fact, two less free parameters since π is normalized and R is scaled to result in unit expected number of changes per unit time at equilibrium).

Parameterizing π. A Markov network (Jordan, 1999) succinctly describes the joint distribution of a set of random variables, given a graph that encodes the dependence structure between these variables. To define π succinctly, we use a Markov network that has the same graph structure as the canonical pathway edges. As shown in Figure 4.2, if U_i is a random variable indicating the presence of pathway gene i, then the Markov network imposes the same graph structure among the U_is as the pathway edges. The joint distribution of this Markov network is calculated as a normalized product of pairwise potentials $f(\cdot,\cdot)$, one per edge of the network. That is, given a realization $u = \{u_i\}$ of the random variables $U = \{U_i\}$, we have $\pi_u = Pr[U = u] \propto \prod_{\text{pathway edges } e=(i,j)} f(u_i, u_j)$.

In this model for π, we make the implicit assumption that an edge between two pathway genes is an indication of their dependent evolution and capture it by the corresponding edge potential $f(\cdot,\cdot)$. This assumption is consistent with our premise outlined in Section 4.1.1. Each edge potential function is a table with three entries or parameters, whose values across all edges are in fact set (tied) to the same three free parameters, $f(0,0), f(1,1)$, and $f(1,0) \doteq f(0,1)$. These free parameters respectively indicate the relative preference of having two characters at the endpoints of an edge to be both absent, both present, or one of them absent and the other present. In a typical scenario of co-evolution, $f(0,0), f(1,1)$ values are larger than $f(0,1)$. Note that tying all edge potential functions to the same function $f(\cdot,\cdot)$ assumes the same strength of co-evolution for all interacting gene pairs, and is hence an approximation to how interacting gene pairs might actually co-evolve. However,

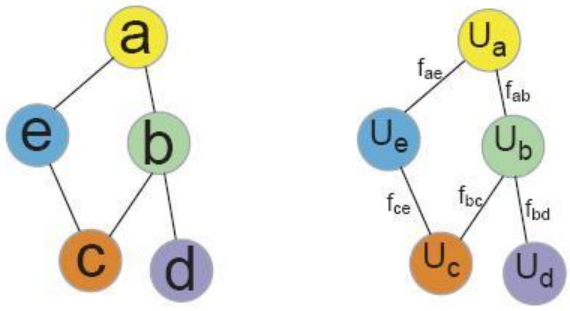

Figure 4.2. A pathway of genes and canonical edges between them (left), and its equivalent Markov network (right). Recall that canonical edges are based on a knowledge of the pathway in a few model organisms. In the Markov network, the U_i random variable indicates the presence of gene i, and f_{ij} is a shorthand for the edge potential $f(U_i, U_j)$. In the typical scenario of co-evolution of interacting genes, $f(0,0)$, $f(1,1)$ values are larger than $f(0,1) \doteq f(1,0)$.

this approximation seems necessary to obtain a meaningful number of parameters that does not increase with the pathway size.

The three free parameters $f(0,0), f(1,1)$, and $f(0,1) \doteq f(1,0)$ can be estimated directly from data without the use of a tree topology, analogous to how equilibrium distribution of nucleotide substitution models are directly estimated from the nucleotide composition of sequences. Thus the parameters of π are considered fixed and constant across tree topologies, and estimated via maximum likelihood (ML) directly from the presence or absence of the k pathway genes in the set of species under study. In detail, we calculate empirical counts of the presence or absence of the endpoints of every edge, use these counts and the above formula to compute the likelihood function $\prod_{\text{species gene contents } u} Pr[U = u]$, and pick the three free parameters that maximize this function (via standard numerical optimization). To obtain non-zero π_u for all u, we added extra unit pseudo-counts to the three empirical counts (Durbin *et al.*, 1998).

We mention a final technical detail about the model. In theory, a pathway could have isolated genes i.e., genes that do not interact with any other pathway gene and hence evolve independently. We factor them into the model via another potential function $g(0), g(1) \doteq 1 - g(0)$, which as before are tied to the same two values across all isolated genes and ML estimated directly from the data (via empirical counts of the presence and absence of all

isolated genes in the set of species, added with extra unit pseudo-counts). Note that the free parameter $g(0)$ is simply the probability that an isolated gene is absent in a species. So to update the above formula for $Pr[U = u]$, we simply multiply the terms in this formula with the probability of presence or absence of every isolated gene.

Parameterizing λ. We start with the simplest assumption that all genes change (are gained/lost) at the same rate, and set this uniform rate without loss of generality to 1 in our experiments (as the rate matrix R is subsequently scaled as mentioned in the beginning of this section).

In a possible extension of this approach, we could use some external (biological) information to partition the genes into a small constant number $c > 1$ of different rate classes and assume that each class j has a specific rate of change r_j. This leaves the model with only c parameters $r_1, \ldots, r_c$, whose numeric values could either be fixed from external information or ML estimated for any given tree topology. This approach is similar to the partition-specific or site-specific rates used in sequence evolution models that allow rate variation among sites (Sullivan and Joyce, 2005). Using discrete Gamma-distributed rates with c rate categories (Yang, 1994) is another popular approach to model rate variation among sites in sequences. However, such an approach would overly complicate our model, as the dependence between character evolutions (gene gains or losses) blows up the time for transition probability calculations by a factor of k^c (as compared to a factor of c for independent site models).

4.2 Phylogenetic Estimation

Having specified our model of pathway evolution in the last section, we are left with using this model to estimate the phylogeny of a pathway present in multiple species. Estimating a phylogeny involves inferring the topology and branch lengths of the tree, which we describe first. We then describe how to assess the confidence value (resampling support) on each branch of the tree using a modified jackknife procedure.

4.2.1 Tree Estimation using the Model

Distance matrix method. Estimating a pathway tree using our model with a maximum likelihood method is conceptually straightforward but computationally intensive. Hence we use a standard distance matrix method, whose inputs are distances between the pathway states in every species pair estimated using our evolutionary model. The output of the method is a tree topology with branch lengths that best fits the input pairwise distances according to the statistically justified least-squares criterion (Felsenstein, 2003).

Our formal estimate of the distance between two pathway states u, v is based on the maximum likelihood (ML) framework. This ML estimate is simply the time t that maximizes the transition probability $[P(t)]_{uv}$ given by our evolutionary model. To estimate the all-pairs distances, we also set beforehand the model parameters (π, λ) as discussed in the previous section. Finally, if multiple species have the same pathway gene content, they are collapsed into a single leaf node of the estimated tree. Although we do not resolve the tree topology between such species, it could be achieved by using sequence information, as detailed in the Discussion.

Implementation pipeline and limitations. The inputs to our phylogenetic estimation method are the pathway gene content in a given set of species, and the canonical edges in the pathway based on known interactions in model organisms. We first estimate the free parameters of π by ML and assume uniform rate for all λs to parameterize our evolutionary model (as explained in Section 4.1.3). Then as mentioned above, we estimate the distances between every species pair in which the pathway is present by ML, and the phylogeny using the least squares distance matrix method FITCH of the Phylip phylogeny inference package (Felsenstein, 1993) with default options. ML estimation of the free parameters of π and all-pairs distances are done via standard numerical optimization methods (specifically quasi-Newton methods in the OPT++ package (Meza, 1994); they are iterative methods whose search for the local maxima of a function is guided by approximations of the function's second-order derivatives (Press *et al.*, 1992)).

To prepare the pathway gene content, we select genes that are experimentally associated with the pathway in model organisms, and search for orthologs of these genes in the genome sequences of a given set of species. These steps are detailed in Supplemental Text B, and their caveats mentioned here. We find orthologs of a gene based on a three-way reciprocal BLAST best hits procedure. This procedures makes simplifying assumptions about orthology to achieve fast running times. But it suffers from some shortcomings including false negative errors due to the stringent three-way requirement, or non-orthologous gene displacements, and false positive errors in the face of events like domain shuffling in multi-domain proteins, domain fusion/fission, sequencing errors or insufficient masking of low-complexity regions in the genome, and large sequence divergence within a protein family (Galperin and Koonin, 1998; Brown and Sjölander, 2006). To partially alleviate some of these shortcomings, we use a variant of the reciprocal best hits procedure and manually inspect the phylogeny of certain protein families (please see Supplemental Text B).

The transition probability calculations require a diagonalization of the rate matrix R, which is done using CLAPACK (Anderson *et al.*, 1999). Because the dimension of R is exponential in k, we are limited to working with small values of k (e.g. $k \leq 13$). However, if the pathway is decomposable into smaller independently evolving sub-pathways or modules based on the graph components of the pathway or other biological information, then the time and memory requirements are greatly reduced. For example, the diagonalization requires $O(2^{3k})$ time for a pathway of k genes, and only $O(2 \cdot 2^{3k/2})$ time if the pathway is decomposable into two independently evolving sub-pathways of $k/2$ genes each.

4.2.2 Tree Estimation with Resampling Supports

Bootstrap or jackknife resampling methods are routinely used to estimate a sequence phylogeny with confidence value (resampling support) on each branch of the tree (Felsenstein, 2003). Our benchmarking results (Section 4.3.1) underscore the importance of estimating a pathway tree with resampling supports on tree branches. Since our model assumes dependency between characters, we cannot use standard bootstrap methods available for independent data. Note that such a method performs random resamplings of the input

data under the assumption that the data comprises independent samples, estimates a tree for each resampled dataset, and computes the fraction of trees that contain a topological feature of interest (e.g. resampling support of a tree branch). We cannot also use existing resampling methods on dependent data, which are studied mainly in the context of stationary, time-series processes (Lahiri, 2003). To suit our Markov network based model of dependent characters, we hence modify the jackknife procedure.

A standard leave-one-out jackknife procedure leaves each character (gene in our case) out of the original input data to obtain several resampled datasets, estimates a tree topology for each such dataset, and obtains a consensus topology of these trees that includes branches supported in at least 50% of the resampled trees (Felsenstein, 2003). This final consensus tree with possible multifurcations (internal nodes with more than three neighbors) is presented along with the resampling supports on each branch (i.e., fraction of resampled trees in which this branch appears). The procedure is implemented in Phylip CONSENSE (with the Majority rule option, which also heuristically resolves some multifurcations). The lengths of the consensus tree branches that best fit the original input data could also be estimated, for instance under the distance matrix framework (implemented in Phylip FITCH, with the same options as in the previous section, and an additional user-tree option set to the consensus tree topology).

In the context of our dependent character model, we need to modify this procedure to update the Markov network over the remaining (non left-out) characters. To do so, we are guided by the concept of an independence-map or *I*-map. A Markov network is an *I*-map for a probability distribution over a set of random variables, if every conditional independence statement implied by the network structure also holds under the distribution (Castillo and Hadi, 2006). From our definition of the joint distribution π over all the characters (Section 4.1.3), it follows that the Markov network in our model is an *I*-map for π. To be consistent, the new Markov network in a jackknife resampling step should similarly be an *I*-map for the marginal distribution of π over the remaining characters. This is achieved simply by removing the left-out node and its incident edges from the original network and adding some minimal extra edges to obtain the new Markov network. These extra edges couple all

neighbors of the left-out node (i.e., they link all pairs of neighbors in the original Markov network of the left-out node). We use this modified, leave-one-out, jackknife procedure to obtain our results.

To increase the number of resampled datasets to obtain possibly more reliable resampling supports, we could also leave more than one gene out in every resampling step. In such a case, the left-out genes in a resampling step are eliminated sequentially in some arbitrary order, where an elimination involves removing a left-out node with its incident edges and adding extra edges to couple their neighbors in the current network (the elimination order of left-out genes won't affect the extra edges added in the final Markov network).

4.3 Results

4.3.1 Simulation Studies on Tunable Pathways

We first benchmark the performance of our method on artificial pathways with tunable number of genes and network structure. This allows us to estimate the error inherent in using gene content or interaction data from a limited number of genes. Given an artificial pathway and a reference phylogenetic tree, the evolution of pathway gene content is simulated along the reference tree according to our substitution model (after assigning the tree root to a gene content drawn from π). To recover an estimate of the phylogenetic tree, the method is then applied to just the data observed at the leaves of the tree (representing the extant species).

We use a family of artificial pathways whose network structures are representative of the typical structure of linear signaling pathways and dense protein complexes. The network structure is tuned via two parameters: number of genes N and edge density K, inspired by the NK-model of tunable fitness landscapes (Kauffman and Levin, 1987). Specifically, a NK pathway is over N genes $g_1, g_2, \ldots, g_N$ such that every gene g_i is connected to each of its K adjacent neighbors $g_{i+1}, g_{i+2}, \ldots, g_{i+K}$ through an edge (ignoring out-of-boundary edges to $g_j, j > N$; see Figure 4.3). To enforce co-evolution of interacting genes,

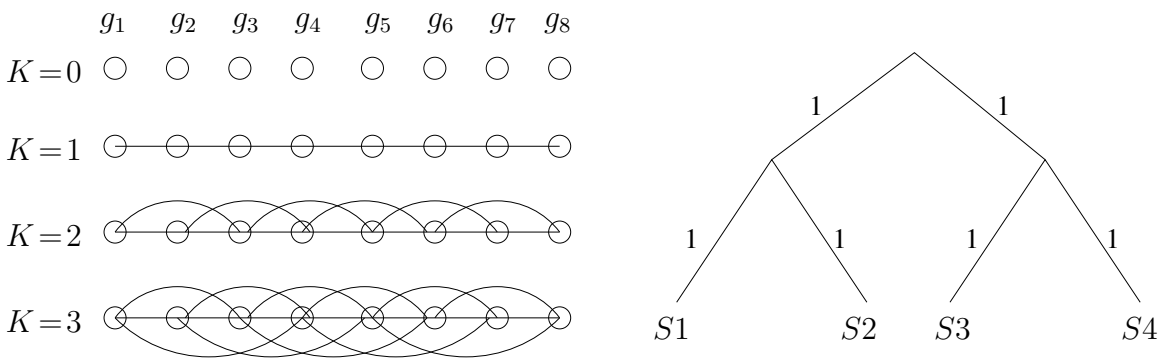

Figure 4.3. The family of artificial pathways inspired by the NK-model is tuned by the number of genes N ($N = 8$ shown) and edge density K ($K = 0, 1, 2, 3$ shown). The reference phylogeny is a complete binary tree with unit branch lengths over n species or leaves ($n = 4$ shown).

No. species (n)	No. genes (N)	Edge density (K)			
		0	1	2	3
4	8	60%	42%	35%	10%
	13	65%	52%	44%	10%
8	8	48%	35%	9%	0%
	13	65%	45%	26%	0%

Table 4.1. Benchmark of tree estimation method using artificial gene content data generated by simulating the evolution of artificial pathways over a reference phylogenetic tree (see Figure 4.3). For each NK pathway, we report here the fraction of 100 simulation trials in which the tree estimated by our method was similar enough (see text for definition) to the reference tree. Note that in each trial, the evolution of the pathway is simulated along the reference tree to generate gene content data at the leaves, and the method is applied to this generated data to estimate the tree.

we use a pairwise potential function: $f(0,0) = 0.35, f(1,1) = 0.35$ that is greater than $f(0,1) \doteq f(1,0) = 0.15$. In the case of $K = 0$, we use $g(0) = g(1) = 0.5$ to capture the evolution of isolated genes.

For evaluation purpose, we declare a tree estimated by our method as *similar enough* to the reference tree if the triplets similarity measure between them is at least 0.5. Given two trees over the same set of n leaves (species), triplets measure quantifies the topological similarity between them as the number of agreeing triplets (sets of three species), normalized by the total number $\binom{n}{3}$ of triplets to yield a value between 0 and 1. A triplet is agreeing between two trees if some ordering of the three species is a *legal triad* in both the trees, where a legal triad refers to an ordered set of three species i, j, k such that the distance

between i, j on the tree is less than or equal to that between i, k and j, k (Hartigan, 1975). If any of these three distances is zero, the triad is not counted as legal, in order to obtain a stringent similarity measure that penalizes some partially resolved trees.

Two observations arise from our benchmark (Table 4.1). First, sparser pathways are better recoverable. The ideal case is when edge density $K = 0$, i.e. when the pathway has no edges. Genes are then independently evolving along the reference tree and hence provide N independent observations (samples) with which to build the tree relationship between the n species. As K increases, pathway genes become more coupled and result in correlated gene contents that are less informative than N independent samples. In fact, we observe that a subset of different species could end up with the same gene content for larger values of K, making it impossible for any gene-content based method to resolve the structure between these species. The worst case is $K = N$, i.e. when all genes are interacting and all are either gained or lost jointly with high probability as the pathway evolves along the reference tree. This level of correlation between the genes effectively reduces the number of samples from N to one. The second observation is that for a given number n of species and a reasonable edge density K, increasing the number of genes N leads to better recovery of the reference tree. This is as expected, since larger N provides more samples that inform the reference tree structure.

4.3.2 Application to Microbial Pathways

These benchmark results suggest that estimated phylogenies of real pathways based on about 13 sparsely interacting genes in 35 or more species would not be fully resolved, and would have multifurcations (internal nodes with more than three neighbors) and different species with the same gene content. The modified jackknife procedure described in Section 4.2.2 can handle this scenario, as it can estimate a partially resolved pathway phylogeny with possible multifurcations, and also measure the confidence value on each resolved tree branch.

We applied our tree estimation method to several microbial pathways, which include

essential metabolic pathways such as glycolysis (Nelson and Cox, 2005, Ch. 14) and the citric acid cycle (Nelson and Cox, 2005, Ch. 16), stress response pathways such as chemotaxis (methyl-processing receptors and enzymatic regulators (Rao *et al.*, 2004)), and cell-cell communication pathways such as quorum sensing (auto-inducers and repressors (Miller and Bassler, 2001)). We trace the evolution of these pathways in a set of 33 representative species, with two species chosen from each major bacterial and archaeal clade (excluding *Planctomycetes* and *Acidobacteria*) to broadly sample all species with sequenced genomes. For other pathways that are not as universally present, species selection is less objective and based on a preliminary clustering of the pathway gene contents. Please see Supplemental Text B for more information on pathway function, node and edge data.

Since research on constructing pathway phylogenies is still in its early stages, there are no "true" pathway phylogenies against which we could compare our estimated pathway trees. A sequence-based species tree (hereafter referred to as the taxonomic tree) was therefore used to provide context. The taxonomic tree is an unrooted ML tree derived from a cleaned and concatenated sequence alignment of 31 universal protein families such as ribosomal and translation-related proteins (Ciccarelli *et al.*, 2006). In addition, we used detailed phenotype data (AH. Singh, DM. Wolf, AP. Arkin, manuscript under review) to evaluate the biological relevance of each phylogenetic branch point.

Discrepancies between phylogenies built for the same set of species but from different source data have been previously recorded. For instance, single gene trees do not always recapitulate taxonomic trees built from universal gene sets. This can be due to lateral gene transfer, gene duplications within a genome, multiple independent gene gains or losses in different branches of the tree, or niche-specific modifications that are not reflected in taxonomic distances (Xie *et al.*, 2003; Bansal and Meyer, 2002; Hooper and Berg, 2003). A pathway phylogeny could also depart significantly from the taxonomic tree. A simple interpretation of this departure in the context of phenotypic data can however predict events such as convergent evolution from similar selective pressures and environmental niches, horizontal gene transfer, and evolution of distinct mechanisms responsible for similar phenotypes (as we show in this section). Although there are sophisticated methods to detect

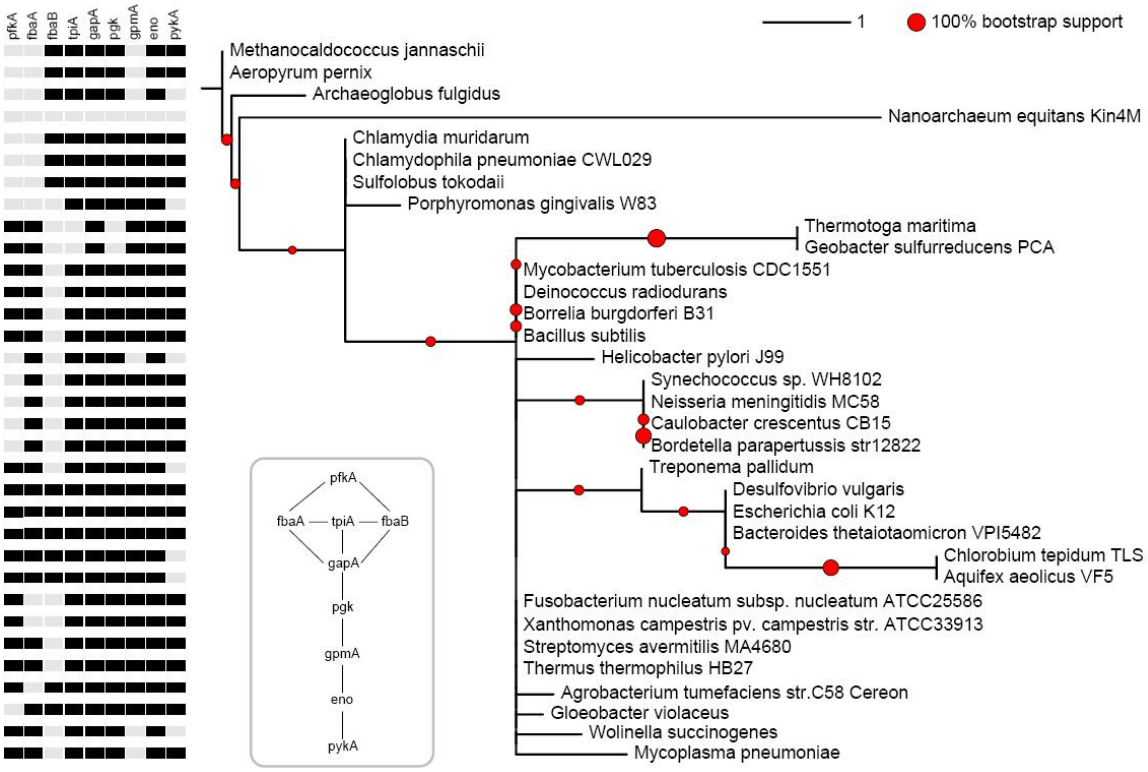

Figure 4.4. Phylogeny of glycolysis pathway for 33 representative species. 9 enzymes and their 11 interactions are used to model the evolution of this pathway. Pathway nodes are enzymes, and edges between nodes indicate that the catalysed reactions between two enzymes involve a common compound. Edges in these pathways are mostly in the form of a linear chain from one enzyme to the next. Node and edge input data are obtained from KEGG (Kanehisa and Goto, 2000). Archaea are well separated from bacteria because of their lack of pfkA and fbaA.

each of these events exclusively, the pathway phylogeny provides a single, concise depiction of several such events in the evolutionary history of the pathway.

Lifestyle-specific gene loss in metabolism

The predicted phylogenies for glycolysis and the citric acid cycle over the 33 representative species are shown in Figures 4.4, 4.5. Although the trees have unresolved multifurcations due to the strong conservation of these metabolic pathways (especially glycolysis) and the small number of query genes, they are nevertheless interpretable if we focus on the resolved clades that have reasonable resampling support. We hence focus on the citric acid

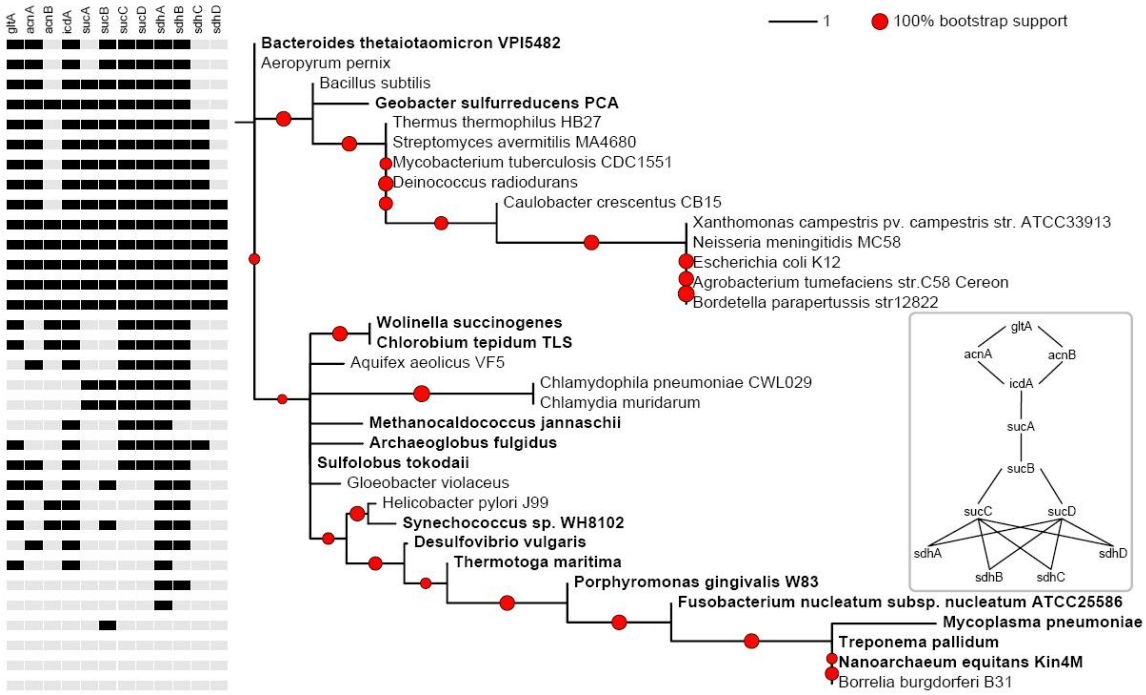

Figure 4.5. Phylogeny of citric acid cycle for 33 representative species. 12 enzymes and their 16 interactions were used to model the evolution of this pathway. Strictly anaerobic species are marked in bold. Pathway nodes are enzymes, and edges between nodes are defined and obtained the same way as for glycolysis. Obligate intracellular symbionts and parasites cluster together as expected. The presence or absence of sucA effectively distinguishes the aerobes from the anaerobes, which run the two halves of the citric acid cycle separately.

tree in this section, which is better resolved than the glycolysis tree. On the glycolysis tree (Figure 4.4), we simply note that the archaeal species are well separated from bacteria.

There are several interesting discrepancies between our predicted citric acid tree (Figure 4.5) and the reference taxonomic tree (triplets similarity of 35% between the two trees). The citric acid tree clusters together species that are separated by large distances in the taxonomic tree. For example, the cluster of four species (*Nanoarchaeum equitans, Borrelia burgdorferi, Treponema pallidum, and Mycoplasma pneumoniae*) belong respectively to the Archaea, Spirochetes (two species), and Firmicutes. They are united, however, in having few or no genes of the citric acid cycle. Phenotypic annotations confirm that they are all obligate intracellular symbionts or parasites[1]: *N. equitans* is an intracellular symbiont of the

[1]Obligate refers to an absolute requirement, so obligate symbionts or parasites do not have an alternate free lifestyle. For survival, an obligate symbiont relies on the organism that it shares a mutually beneficial relation with. An obligate parasite similarly relies on the host organism for survival, causing some harm to the host in the process.

marine archaeon *Igniococcus*, *B. burgdorferi* a parasite of ticks and the causative agent of Lyme disease, *T. pallidum* a parasite causing syphilis, and *M. pneumoniae* another human pathogen causing pneumonia. A parasitic lifestyle in a nutrient-rich host environment has been suggested to cause the loss of metabolic genes (Lawrence and Hendrickson, 2005); our phylogeny estimation method detects and illustrates this in an automated way.

Progressing from reduced genomes to metabolically complete genomes on the citric acid cycle tree reveals a gradual gain/loss of isoenzymes (multiple forms of an enzyme coded for by different genes), co-evolution of subunits of multi-enzyme complexes, and phenotypic association of evolutionarily critical enzymes. We first note that the co-evolution between isoenzyme pairs is much weaker than that between interacting subunits of the same enzyme complex, thereby supporting the use of canonical pathway edges in building a co-evolution model. For example, 12 of 33 species have aconitase AcnA but not its isoenzyme AcnB, 4 species have AcnB but not AcnA, and 11 species have neither (Figure 4.5). In contrast, the E1 and E2 subunits of the 2-oxoglutarate dehydrogenase complex (SucAB) are both present or both absent in all but 5 of the 33 species.

Moving onto α-ketoglutarate dehydrogenase (SucA), we see a striking phenotypic association – of the 19 species lacking SucA, 14 are obligate anaerobes that branch deeply on the tree (marked in bold on Figure 4.5). Since the citric acid cycle only operates in aerobic cells, anaerobes run the four-carbon part of the pathway (succinate to oxaloacetate) in reverse, presumably for reductive biosynthesis (Michal, 1999; Huynen *et al.*, 1999). It is presumed that before advent of oxygen on the earth's surface, early hyperthermophilic bacteria and archaea had both segments of the citric acid cycle, with each operating separately as a linear pathway (succinate to oxaloacetate, and oxaloacetate to α-ketoglutarate). In aerobic species, the appearance of α-ketoglutarate dehydrogenase (SucA) then unified the two linear pathways into a cycle (LaNoue, 2001). Finally, the clade of species with a complete complement of enzymes (*X. campestris, N. meningitidis, E. coli, A. tumefaciens, B. parapertussis*) is both the most metabolically and phenotypically versatile of the species set, since biochemically equivalent enzymes could be regulated differently according to different physiological conditions (Serres and Riley, 2006).

Evolution of chemotaxis modulation

Chemotaxis is a well-studied signal transduction pathway that bacteria use to sense changes in the chemical composition of their environment and move accordingly. This is achieved by transducing signals from methyl-accepting receptor complexes (McpABC, TlpABC, Tar, Tap), located at the poles of the cell, via the CheY messenger protein, to flagellar-motor complexes evenly distributed around the cell. An intricate signal cascade (CheABCDRVZ) modulates the phosphorylated form of the messenger protein CheY and steers the cell in the appropriate direction (Sonenshein *et al.*, 2002, Ch. 31). In *E. coli* and *B. subtilis*, the chemotaxis network comprises approximately 60 genes. Roughly half these genes code for the flagellar apparatus, and another third for membrane-anchored receptors that bind specific extracellular ligands. We focus on the receptors and signal transduction modules of the network, and one flagellar protein, FliM, that is directly regulated by CheY.

There is a significant amount of variation in chemotaxis gene content even among motile species, and the tree we predict (Figure 4.6) groups this variation into three major classes, which reflect the knowledge of chemotaxis in model organisms. The main cross-species differences are in the distribution of regulators that modulate CheW activity: CheC (present in 24/88 species), CheD (44/88 species), and CheV (37/88 species). CheZ, which deactivates the master regulator CheY, is also not uniformly conserved (35/88 species). Thus, whereas the major ligand-binding (Mcp, Tlp) and two-component system (CheAY) of chemotaxis is conserved in all motile species, the "modulation" proteins are not. This difference has been documented in detail for *E. coli* (which lacks CheCDV), and *B. subtilis* (which lacks CheZ) (Rao *et al.*, 2004). We confirm that our pathway tree groups 73 of the 88 species into three broad gene content classes: "*E. coli*-like" content lacking one of CheCDV but having CheZ (the clade of 31 species containing *E. coli* and *C. violaceum*), "*B. subtilis*-like" content lacking CheZ but having CheCD (the clade of 21 species containing *B. subtilis* and *Borrelia spp.*, with the latter having only CheD), and a third gene content class lacking both CheZ and CheCD (the clade of 21 species containing *M. loti* and *D. radiodurans*).

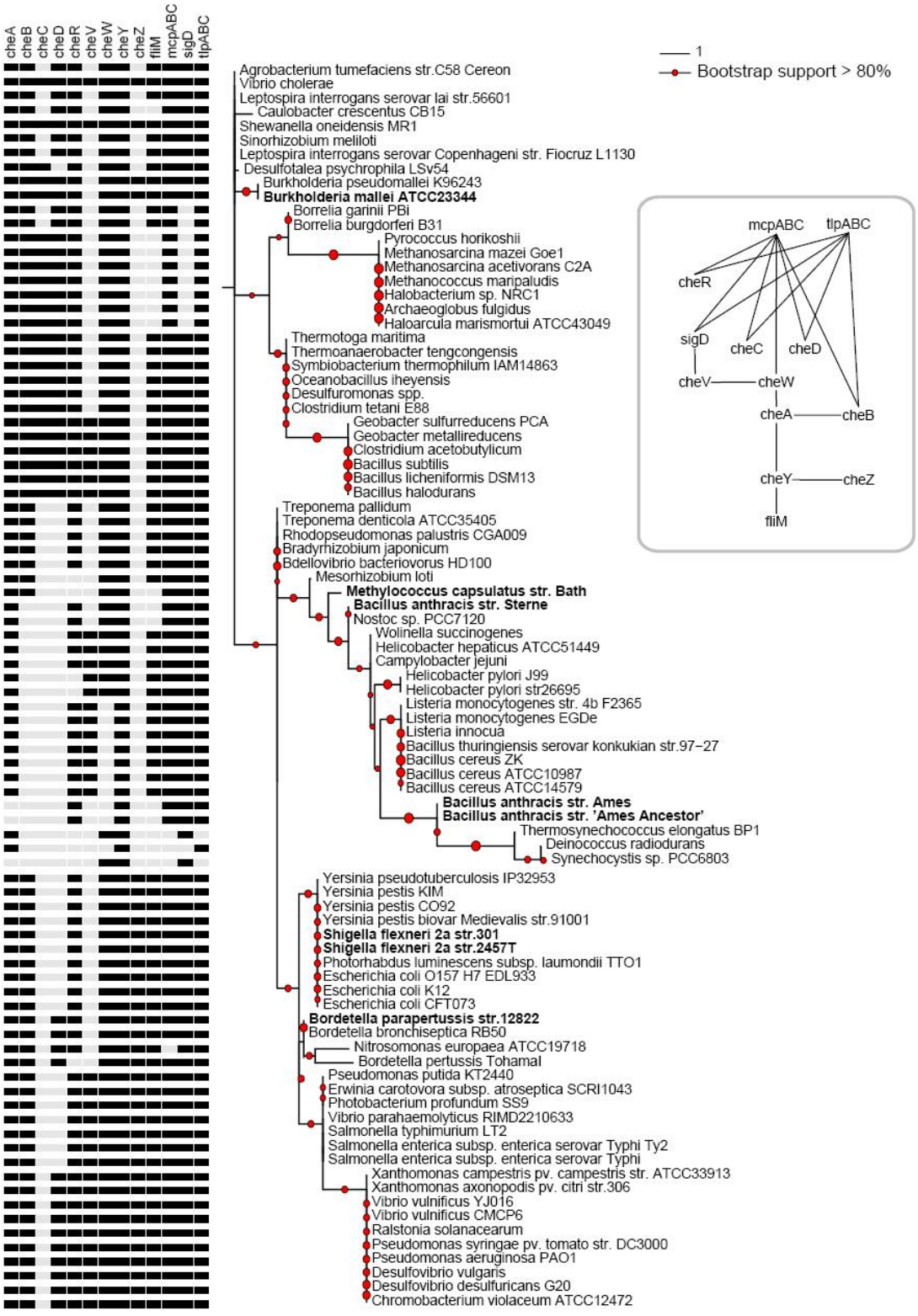

Figure 4.6. Phylogeny of chemotaxis for 88 selected species, built using 13 genes and their 18 interactions. Non-motile species are marked in bold.

Observe that non-motile species appear in the same clade as motile species from the same genus due to identical gene content. In some instances where this occurs, a non-motile obligate pathogen such as *Bordetella spp.* or *S. flexneri* clusters with a motile free-living species such as *E. coli*. This could be explained by different evolutionary scenarios. The last common ancestor could have been motile, and the loss of the phenotype occurred after speciation along one branch, where the genes were retained as pseudogenes or inactive copies. Alternatively, motility could have been preserved along one branch, while the genes in the other branch underwent divergent evolution and were recruited to other regulatory pathways. The former explanation is partly supported by evidence that flagellar orthologs in *Bordetella parapertussis* are inactivated pseudogenes (Parkhill *et al.*, 2003).

Combinatorial evolution of quorum sensing systems

Quorum sensing is the ability of individual cells to regulate gene expression in response to variations in population density. Bacteria achieve this by producing signaling molecules called autoinducers that are secreted into the extracellular medium. When a threshold level of autoinducer is detected, it leads to a change in gene expression. In this manner, bacteria regulate a variety of biological functions such as symbiosis, virulence, antibiotic production, motility, sporulation, and biofilm formation. There are at least four alternate mechanisms to secrete and detect autoinducers (Miller and Bassler, 2001): (i) the best-studied LuxI/LuxR-type quorum sensing system of Gram-negative bacteria, with variations of this two-component system present in *V. fischeri*, *P. aeruginosa*, *A. tumefaciens*, and *E. carotovora*; (ii) the peptide-mediated quorum sensing in Gram-positive bacteria, with variations among *B. subtilis*, *S. pneumoniae*, and *S. aureus* (Storz and Hengge-Aronis, 2000); (iii) the multi-channel lux circuit of *V. harveyi* (Waters and Bassler, 2006); and (iv) the unique amino acid secretion system of *M. xanthus* (Kuspa *et al.*, 1992). For this analysis, we used 35 representative genes from three well-studied quorum sensing systems in *V. harveyi*, *B. subtilis*, and *A. tumefaciens*.

Since the quorum sensing tree was derived from genes in three species with different quorum sensing mechanisms, we first verified that species with similar mechanisms group

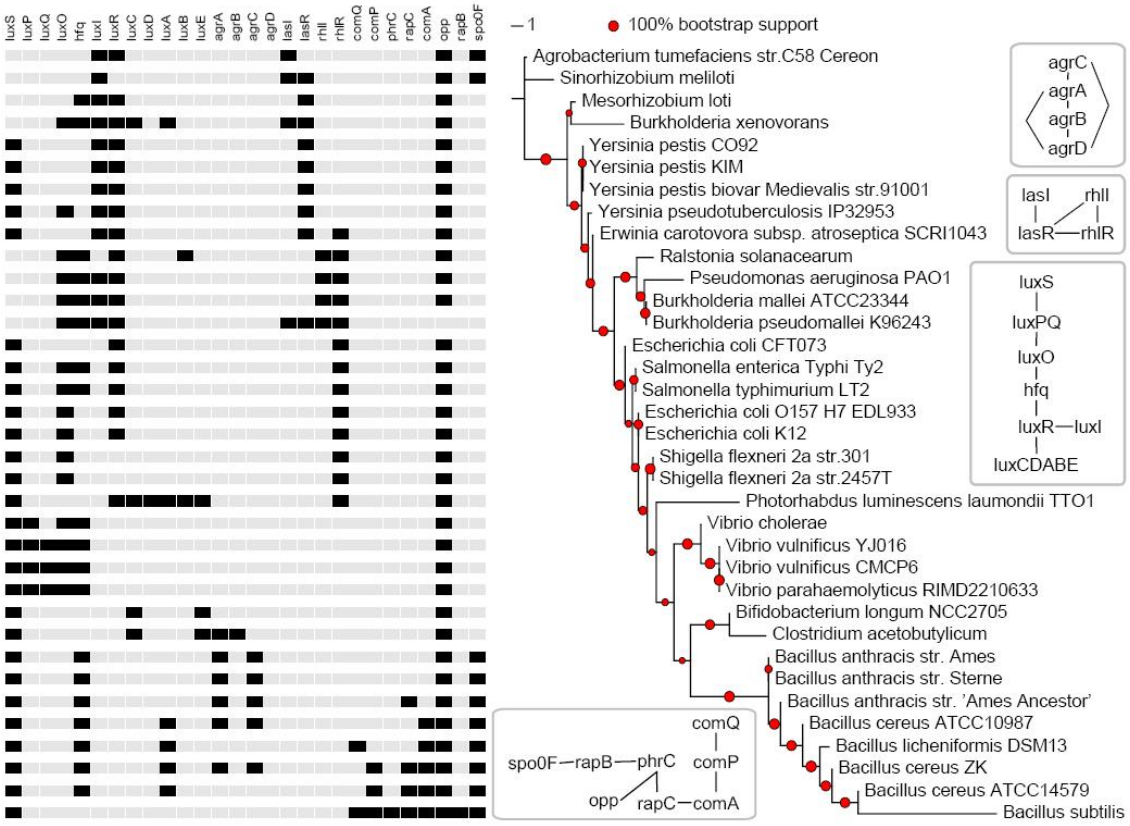

Figure 4.7. Phylogeny of quorum sensing for 35 selected species built from gene and interaction data on multiple quorum sensing mechanisms. A total of 28 genes and their 28 interactions are used. In the interest of space, the pathway shows luxC, luxD, luxA, luxB and luxE genes as luxCDABE. Note that our analysis treats these genes as separate nodes with each having a separate edge to luxR, and no edges amongst themselves. The analysis similarly treats luxP and luxQ separately, though the pathway shows them as luxPQ.

together. This is indeed the case (Figure 4.7). For instance, the tree clusters together the *Vibrio* species, which all use LuxS to produce the autoinducer, LuxPQ to detect threshold autoinducer levels, and LuxO and Hfq as intermediate relays in the regulation of target genes. Also clustered together are species sharing the paralogous two-component systems RhlI/RhlR and LasI/LasR systems, in which RhlI or LasI produce the autoinducer, and RhlR or LasR detect it (Miller and Bassler, 2001). This clade includes *Pseudomonas*, *Burkholderia*, and *Ralstonia* species.

Interestingly, *B. subtilis* also uses a homolog of LuxS for autoinducer secretion as in *Vibrio*, and a probable Lsr-like system instead of LuxPQ for signal detection, in addition to the CSF/ComX systems. This observation made in a recent study is along the lines

of two similar observations in earlier studies that suggest a LuxS/Lsr-like quorum sensing mechanism in *B. anthracis* and *B. cereus* (Lombardia *et al.*, 2006). Moreover, since the pathway tree places *B. licheniformis* close to *B. subtilis* and *B. cereus* in the same clade, we predict that it possesses a LuxS/Lsr-like system as well.

4.4 Discussion

In this work, we propose a probabilistic model of pathway evolution that is tuned to published genomic and pathway data from almost a hundred lineages of bacteria and archaea. The model is tractable with gene presence or absence defining the pathway state, and canonical interactions indicating the static dependence between the co-evolving genes. We apply this model to estimate the phylogeny of several conserved pathways in metabolism, stress response, and intercellular communication. Our model is both flexible and applicable beyond phylogenetic estimation. If the true pathway phylogeny was known, our model could be used to test evolutionary hypotheses about the dependence constraints on pathway evolution. Also, the general k-character co-evolution model can be used in an entirely different setting to make testable predictions. For example, it could be adapted to study co-evolution of domain content in a protein, and used with an inference procedure to make function predictions. This would be a refined version of SIFTER (Engelhardt *et al.*, 2005), a Bayesian phylogenomic model for function prediction. The refinement comes from imposing strong dependence between the evolving domains throughout evolutionary time, instead of just during ancestral branching points as done in SIFTER.

Although tree topology is currently not resolved among a group of species with the same gene content, it could be resolved using sequence information, as follows. A rooted phylogeny is first estimated for each such group of collapsed species S_L at a leaf L by choosing an outgroup species that contains as many of the genes present in S_L as possible. This can be done by a combined analysis (Yang, 1996, mixed data model) that uses sequence data of all pathway genes common to S_L and the outgroup. The original estimated tree is then refined by attaching at leaf L the rooted phylogeny between the S_L species. This

technique of reconstructing phylogenies by incorporating events at different spatial and temporal scales (i.e., "macro"-level events such as gain/loss of genes and "micro"-level events such as nucleotide sequence mutations) is similar to previously described techniques (Durand *et al.*, 2005).

One limitation of our model is its exponential dependence on the number of genes k. Large k are needed to properly resolve the evolutionary relationship between the pathway in different species. To circumvent this partially, sequence information could be incorporated as discussed above to resolve collapsed branches caused by identical gene content. Nevertheless, allowing large k would be preferable. One solution would be to decompose the pathway into small ($k \leq 13$) independent sub-pathways. Another solution would be to exploit the sparsity of R by using Taylor's expansion of matrix exponentials, and it might permit $k \leq 18$, for example. To handle even larger k, basic changes in modeling are required. One approach to pursue would be to use a discrete time approximation of the continuous time Markov chain in our model, and exploit the dependence structure of the resultant discrete time model. Another approach would be to compromise and use models similar to phylo-HMMs (Siepel and Haussler, 2004) that allow only weak dependency between characters or sites by only allowing correlations between rates. Modeling strong dependence between large-scale number of characters or sites remains an active open area of research (Pedersen and Jensen, 2001; Robinson *et al.*, 2003).

We have focused mainly on pathways in closely related bacterial species, which makes our assumption of a static dependence structure between evolving nodes more justifiable than in the case of distantly related species. It also simplifies our task of establishing one-to-one correspondences between pathway nodes in different species; these pathway alignments are inputs to the model. But we note that our pathway alignment method (described in Supplemental Text B) is not free of caveats even across closely related species (see Section 4.2.1; for example, alignment errors could result from not explicitly handling domain shuffling and fusion/fission events). For the case of distantly related species, our method works conceptually. Still, over large evolutionary times, gene duplication and other mutation events might lead to radical changes in pathway structure. This makes pathway

alignment difficult, and also prevents the evolutionary model from being faithful to the structural changes. Although the problem of simultaneously studying pathway alignment and evolution has been tackled by combinatorial graph-based algorithms (Heymans and Singh, 2003), it has not been handled by tractable probabilistic models.

We present results on microbial pathways, for which there is a wealth of genetic and genomic data. Our results show that a pathway phylogeny can provide a concise depiction of disparate events in the evolution of a pathway, and that studying discrepancies between gene sequence data, pathway phylogeny, and phenotype data is an effective way to infer pathway-wide evolutionary hypotheses. Traditionally, gene sequences are used to study phylogeny, and ecological methods are used to study how selection (adaptation) acts on phenotype. Building and analysing pathway phylogenies provides a bridge between these two methods, because pathways are associated with specific cellular phenotypes but can also be viewed as coherently evolving unit of genes and interactions.

Chapter 5

Future Work

We presented new computational methods for the interrelated problems of alignment and phylogeny of biological networks. We applied the methods to study cross-species conservation of protein interaction networks, and evolution of bacterial and archaeal pathways involved in cellular metabolism and signaling. As highlighted in Section 1.4, this thesis significantly advances existing methodologies: the *Match-and-Split* graph-matching algorithm has provable guarantees unlike earlier heuristic approaches, and our probabilistic model with explicit assumptions about pathway evolution is in contrast to discrete similarity based approaches to estimate pathway phylogenies. The previous chapters described at length the design of our methods, their application on cross-species sequence and interaction data, and interpretation of the results. This chapter concludes with the research problems and future areas that the thesis opens up.

The computational problems to compare graphs in Chapter 3 could be extended in several directions. The extensions would increase the applicability of graph matching not just in the field of biological network comparison, but also in other fields such as computer vision.

- The *Match-and-Split* algorithm is provably efficient for a family of local matching and connectivity criteria, and expanding this family to obtain a more versatile algorithm results in many interesting open questions. For example, extending the problem from

monotone local matching criterion to a non-monotone criterion makes it NP-hard, and the existence of approximation algorithms remains open. Another problem which is similarly NP-hard and whose approximability is unknown concerns the search for the largest connected and matched subgraph pair between two input graphs. A subgraph pair is matched if there exists a bipartite matching between the nodes in the two subgraphs over the node similarity edges. Note that similar subgraph pairs returned by *Match-and-Split* could exhibit many-many node similarity relations, and need not be matched in general.

- Most of graph matching focuses on finding similar subgraphs between input graphs. Searching instead for dissimilar subgraphs between graphs is informative too and opens up new directions in comparison of cross-tissue or cross-condition (disease vs. healthy) interaction networks. If we assume some similarity relationship between nodes in the input graphs, then we can define dissimilar subgraphs, one from each input graph, as subgraphs over related set of nodes but exhibiting different edge densities or dissimilar interaction patterns between the input graphs.

- Another direction is to search for algorithms with better running time guarantees than the *Match-and-Split* algorithm for some commonly used local matching criterion. For instance, it is open if the recursive *Match-and-Split* algorithm could be made recursion-less and more efficient for the 1-similar paths criterion.

Modeling pathway evolution along a phylogeny is a nascent research area with many open questions too. We mention here a few future directions of our pathway phylogeny work of Chapter 4 (see also Section 4.4).

- Modeling strong dependence (co-evolution) between large-scale number of characters or sites remains active and open (Pedersen and Jensen, 2001; Robinson *et al.*, 2003). This would address a limitation of our model viz., its exponential dependence on the number of genes k.

- Our results indicate discrepancies between pathway trees and accepted sequence-based species trees. Statistical significance of such discrepancies could be measured against

a null model of discrepancies that arise purely from noise in the input data (e.g. due to small sample size) or in the tree reconstruction procedure. Development of statistical tests that measure this significance could reliably inform us about environmental niches that adapt certain pathways faster than sequences of universal protein families over evolutionary time.

- Testable predictions of cellular phenotypes could benefit from pathway evolution models. As an example, we could model the presence or absence of a pathway-related phenotype along with the pathway gene content using our co-evolution model, and use it with a pathway phylogeny and phenotypic data available for some species to predict the phenotype in other species.

- Obtaining accurate gene content of a pathway is a problem orthologonal to but important for the accurate reconstruction of the pathway phylogeny. In future, we would explore more systematic alternatives such as phylogenomic analysis (Eisen, 1998) to our current orthology finding method based on a reciprocal best hits criterion (Section 4.2.1).

The computational framework in this thesis could be extended to study a largely uncharted research area concerning the evolution of transcriptional regulation. Regulation of a set of genes is as important as the biochemical functions the genes code for, and increasing evidence suggests that variation in regulatory sequences in the genome, besides coding sequences, are major contributors of phenotypic evolution (Wray *et al.*, 2003). This thesis presented cross-species analysis of biological networks that mostly capture the function of a set of genes (i.e., protein interaction, metabolic or signaling networks), however our flexible algorithm framework could potentially be extended to compare networks of gene regulatory interactions inferred from expression data. Gene regulatory networks are useful platforms to examine the evolution of transcriptional regulation (Hinman *et al.*, 2003), since they provide a causal, mechanistic representation of the regulatory programs encoded in the genome.

Bibliography

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P., 2002. *Molecular Biology of the Cell*. Garland Science.

Alm, E. J., Huang, K. H., Price, M. N., Koche, R. P., Keller, K., Dubchak, I. L., and Arkin, A. P., 2005. The MicrobesOnline Web site for comparative genomics. *Genome Research* 15(7), 1015–22.

Alon, N., Yuster, R., and Zwick, U., 1995. Color-coding. *Journal of the ACM* 42(4), 844–856.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Croz, J. D., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D., 1999. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 3rd edition.

Apweiler, R., Bairoch, A., Wu, C., Barker, W., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., and *et al.*, 2004. UniProt: the Universal Protein Knowledgebase. *Nucleic Acids Research* 32, D115–D119.

Aravind, L., Walker, D. R., and Koonin, E. V., 1999. Conserved domains in DNA repair proteins and evolution of repair systems. *Nucleic Acids Research* 27(5), 1223–42.

Baase, S., 1991. *Computer algorithms : Introduction to design and analysis*. Addison-Wesley, 2nd edition.

Bader, G. and Hogue, C., 2003. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* 4, 2.

Bansal, A. K. and Meyer, T. E., 2002. Evolutionary analysis by whole-genome comparisons. *Journal of Bacteriology* 184(8), 2260–2272.

Barker, D. and Pagel, M., 2005. Predicting functional gene links from phylogenetic-statistical analyses of whole genomes. *PLoS Computational Biology* 1(1), e3.

Batzoglou, S., 2005. The many faces of sequence alignment. *Briefings in Bioinformatics* 6(1), 6–22.

Berg, J. and Lassig, M., 2004. Local graph alignment and motif search in biological networks. *Proceedings of the National Academy of Sciences* 101(41), 14689–14694.

Berg, J. and Lassig, M., 2006. Cross-species analysis of biological networks by Bayesian alignment. *Proceedings of the National Academy of Sciences* 103(29), 10967–10972.

Boldogkoi, Z., 2004. Gene network polymorphism is the raw material of natural selection: the selfish gene network hypothesis. *Journal of Molecular Evolution* 59(3), 340–357.

Bork, P., Jensen, L., von Mering, C., Ramani, A., Lee, I., and Marcotte, E., 2004. Protein interaction networks from yeast to human. *Current Opinion in Structural Biology* 14(3), 292–299.

Boyle, E., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J., and Sherlock, G., 2004. GO::TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics* 20(18), 3710–3715.

Brohée, S. and van Helden, J., 2006. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics* 7, 488.

Brown, D. and Sjölander, K., 2006. Functional classification using phylogenomic inference. *PLoS Computational Biology* 2(6), e77.

Bunke, H., 1999. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 21(9), 917–922.

Burset, M. and Guigo, R., 1996. Evaluation of gene structure prediction programs. *Genomics* 34(3), 353–367.

Castillo, E. and Hadi, A. S., 2006. Markov Networks. *Encyclopedia of Statistical Sciences* 7, 4535–4546.

Chor, B. and Tuller, T., 2006. Biological networks: comparison, conservation, and evolutionary trees. *Proc. Intl. Conf. on Research in Computational Molecular Biology* 3909, 30–44.

Ciccarelli, F. D., Doerks, T., von Mering, C., Creevey, C., Snel, B., and Bork, P., 2006. Toward automatic reconstruction of a highly resolved tree of life. *Science* 311(5765), 1283–1287.

Deng, M., Sun, F., and Chen, T., 2003. Assessment of the reliability of protein-protein interactions and protein function prediction. In *Pacific Symposium on Biocomputing*, 140–151.

Dolezal, P., Likic, V., Tachezy, J., and Lithgow, T., 2006. Evolution of the molecular machines for protein import into mitochondria. *Science* 313(5785), 314–318.

Durand, D., Halldórsson, B. V., and Vernot, B., 2005. *A Hybrid Micro-Macroevolutionary Approach to Gene Tree Reconstruction*, volume 3500. Springer-Verlag, Berlin.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G., 1998. *Biological Sequence Analysis: Probabilistic Models of Protein and Nucleic Acids*. Cambridge University Press, UK.

Dutkowski, J. and Tiuryn, J., 2007. Identification of functional modules from conserved ancestral protein-protein interactions. *Bioinformatics* 23, 149–158.

Edgar, R. C., 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucl. Acids Res.* 32(5), 1792–1797.

Eisen, J. and Hanawalt, P., 1999. A phylogenomic study of dna repair genes, proteins, and processes. *Mutation Research/DNA Repair* 435(3), 171–213.

Eisen, J. A., 1998. Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome Research* 8(3), 163–167.

Engelhardt, B. E., Muratore, K., Brenner, S. E., and Jordan, M. I., 2005. Protein molecular function prediction by Bayesian phylogenomics. *PLoS Computational Biology* 1(5), 432–445.

Felsenstein, J., 1993. PHYLIP – Phylogeny Inference Package version 3.2. *Cladistics* 5, 164–166.

Felsenstein, J., 2003. *Inferring Phylogenies*. Sinauer Associates, Inc., Sunderland, MA.

Flannick, J., Novak, A., Srinivasan, B., McAdams, H., and Batzoglou, S., 2006. Græmlin: General and robust alignment of multiple large interaction networks. *Genome Research* 16(9), 1169–1181.

Forst, C. and Schulten, K., 2001. Phylogenetic analysis of metabolic pathways. *Journal of Molecular Evolution* 52(6), 471–489.

Galperin, M. Y. and Koonin, E. V., 1998. Sources of systematic error in functional annotation of genomes: domain rearrangement, non-orthologous gene displacement and operon disruption. *In Silico Biology* 1(1), 55–67.

Garey, M. and Johnson, D., 1979. *Computers and intractability: A guide to the theory of NP-completeness*. Freeman, New York.

Gavin, A.-C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., Rau, C., Jensen, L., Bastuck, S., Dumpelfeld, B., and *et al.*, 2006. Proteome survey reveals modularity of the yeast cell machinery. *Nature* 440, 631–636.

Gavin, A.-C., Bösche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J. M., Michon, A.-M., Cruciat, C.-M., and *et al.*, 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* 415(6868), 141–147.

Giot, L., Bader, J. S., Brouwer, C., Chaudhuri, A., Kuang, B., Li, Y., Hao, Y. L., Ooi, C. E., Godwin, B., Vitols, E., and *et al.*, 2003. A protein interaction map of Drosophila melanogaster. *Science* 302(5651), 1727–1736.

Girvan, M. and Newman, M., 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99(12), 7821–7826.

Goldman, N. and Yang, Z., 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Molecular Biology and Evolution* 11(5), 725–736.

Grebe, T. and Stock, J., 1999. The histidine protein kinase superfamily. *Advances in Microbial Physiology* 41, 139–227.

Guindon, S. and Gascuel, O., 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Systematic Biology* 52(5), 696–704.

Hartigan, J. A., 1975. *Clustering Algorithms*. John Wiley, New York.

Hartwell, L., Hopfield, J., Leibler, S., and Murray, A., 1999. From molecular to modular cell biology. *Nature* 402(6761), C47–C52.

Heymans, M. and Singh, A. K., 2003. Deriving phylogenetic trees from the similarity analysis of metabolic pathways. *Bioinformatics* 19 Suppl 1, i138–46.

Hinman, V. F., Nguyen, A. T., Cameron, R. A., and Davidson, E. H., 2003. Developmental gene regulatory network architecture across 500 million years of echinoderm evolution. *Proceedings of the National Academy of Sciences* 100(23), 13356–13361.

Hirsh, E. and Sharan, R., 2007. Identification of conserved protein complexes based on a model of protein network evolution. *Bioinformatics* 23, 170–176.

Ho, Y., Gruhler, A., Heilbut, A., Bader, G. D., Moore, L., Adams, S.-L., Millar, A., Taylor, P., Bennett, K., Boutilier, K., and *et al.*, 2002. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature* 415(6868), 180–183.

Hong, E., Balakrishnan, R., Christie, K., Costanzo, M., Dwight, S., Engel, S., Fisk, D., Hirschman, J., Livstone, M., Nash, R., and *et al.*, 2006. Saccharomyces Genome Database (Aug 2006). Http://www.yeastgenome.org/.

Hooper, S. D. and Berg, O. G., 2003. On the nature of gene innovation: duplication patterns in microbial genomes. *Molecular Biology and Evolution* 20(6), 945–954.

Huynen, M. A., Dandekar, T., and Bork, P., 1999. Variation and evolution of the citric-acid cycle: a genomic perspective. *Trends in Microbiology* 7(7), 281–291.

Hwang, W., Cho, Y.-R., Zhang, A., and Ramanathan, M., 2006. A novel functional module detection algorithm for protein-protein interaction networks. *Algorithms for Molecular Biology* 1, 24.

Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., and Sakaki, Y., 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences* 98(8), 4569–4574.

Jordan, M. I., 1999. *Learning in Graphical Models*. MIT Press, Cambridge, MA.

Kanehisa, M. and Goto, S., 2000. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* 28(1), 27–30.

Kauffman, S. and Levin, S., 1987. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology* 128(1), 11–45.

Kelley, B., Sharan, R., Karp, R., Sittler, T., Root, D., Stockwell, B., and Ideker, T., 2003. Conserved Pathways within Bacteria and Yeast as revealed by Global Protein Network Alignment. *Proceedings of the National Academy of Sciences* 100(20), 11394–11399.

Kelley, R. and Ideker, T., 2005. Systematic interpretation of genetic interactions using protein networks. *Nature Biotechnology* 23, 561–566.

Kimbrell, D. and Beutler, B., 2001. The evolution and genetics of innate immunity. *Nature Reviews Genetics* 2(4), 256–267.

King, A., Przulj, N., and Jurisica, I., 2004. Protein complex prediction via cost-based clustering. *Bioinformatics* 20, 3013–3020.

Kiritchenko, S., Matwin, S., and Famili, A., 2005. Functional annotation of genes using hierarchical text categorization. BioLINK SIG: Linking Literature, Information and Knowledge for Biology.

Koyuturk, M., Grama, A., and Szpankowski, W., 2005. Pairwise local alignment of protein interaction networks guided by models of evolution. In *Proc. 9th Intl. Conf. on Research in Computational Molecular Biology*, 48–65.

Koyuturk, M., Kim, Y., Topkara, U., Subramaniam, S., Szpankowski, W., and Grama, A., 2006. Pairwise alignment of protein interaction networks. *Journal of Computational Biology* 13(2), 182–199.

Krogan, N., Cagney, G., Yu, H., Zhong, G., Guo, X., Ignatchenko, A., Li, J., Pu, S., Datta, N., Tikuisis, A., and *et al.*, 2006. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature* 440, 637–643.

Kuspa, A., Plamann, L., and Kaiser, D., 1992. A-signalling and the cell density requirement for *Myxococcus xanthus* development. *Journal of Bacteriology* 174(22), 7360–7369.

Lahiri, S., 2003. *Resampling Methods for Dependent Data*. Springer-Verlag, 1st edition.

LaNoue, K. F., 2001. Citric acid cycle. In *Encyclopedia of Life Sciences*. John Wiley, Chichester.

Lawrence, J. and Hendrickson, H., 2005. Genome evolution in bacteria: order beneath chaos. *Current Opinion in Microbiology* 8(5), 572–578.

Lenski, R., Barrick, J., and Ofria, C., 2006. Balancing robustness and evolvability. *PLoS Biology* 4(12), e428.

Li, S., Armstrong, C. M., Bertin, N., Ge, H., Milstein, S., Boxem, M., Vidalain, P.-O., Han, J.-D. J., Chesneau, A., Hao, T., and *et al.*, 2004. A Map of the Interactome Network of the Metazoan C. elegans. *Science* 303(5657), 540–543.

Lombardia, E., Rovetto, A., Arabolaza, A., and Grau, R., 2006. A LuxS-dependent cell-to-cell language regulates social behavior and development in *Bacillus subtilis*. *Journal of Bacteriology* 188(12), 4442–4452.

Matthews, L., Vaglio, P., Reboul, J., Ge, H., Davis, B., Vincent, J. G. S., and Vidal, M., 2001. Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or interologs. *Genome Research* 11(12), 2120–2126.

Mewes, H., Amid, C., Arnold, R., Frishman, D., Guldener, U., Mannhaupt, G., Munsterkotter, M., Pagel, P., Strack, N., Stumpflen, V., and *et al.*, 2004. MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Research* 32(Database issue), D41–D44.

Meza, J., 1994. OPT++: An Object-Oriented Class Library for Nonlinear Optimization. Technical Report SAND94-8225, Sandia National Laboratory, Albuquerque, NM. `http://csmr.ca.sandia.gov/opt++`.

Michal, G., 1999. *Biochemical Pathways: An Atlas of Biochemistry and Molecular Biology*. Wiley, New York.

Miller, M. B. and Bassler, B. L., 2001. Quorum sensing in bacteria. *Annu Rev Microbiol* 55(1), 165–199.

Nelson, D. L. and Cox, M. M., 2005. *Lehninger Principles of Biochemistry*. W.H. Freeman, New York, NY, 4th edition.

Ogata, H., Fujibuchi, W., Goto, S., and Kanehisa, M., 2000. A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Research* 28, 4021–4028.

Parkhill, J., Sebaihia, M., Preston, A., Murphy, L. D., Thomson, N., Harris, D. E., Holden, M. T. G., Churcher, C. M., Bentley, S. D., Mungall, K. L., and *et al.*, 2003. Comparative analysis of the genome sequences of *Bordetella pertussis*, *Bordetella parapertussis* and *Bordetella bronchiseptica*. *Nature Genetics* 35(1), 32–40.

Pedersen, A. and Jensen, J., 2001. A dependent-rates model and an MCMC-based methodology for the maximum-likelihood analysis of sequences with overlapping reading frames. *Molecular Biology and Evolution* 18(5), 763–776.

Pereira-Leal, J., Enright, A., and Ouzounis, C., 2004. Detection of functional modules from protein interaction networks. *Proteins* 54, 49–57.

Peri, S., Navarro, J., Amanchy, R., Kristiansen, T., Jonnalagadda, C., Surendranath, V., Niranjan, V., Muthusamy, B., Gandhi, T., Gronborg, M., and *et al.*, 2003. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Research* 13(10), 2363–2371.

Pinter, R., Rokhlenko, O., Tsur, D., and Ziv-Ukelson, M., 2004. Approximate labelled subtree homeomorphism. *Combinatorial Pattern Matching* 59–73.

Pinter, R., Rokhlenko, O., Yeger-Lotem, E., and Ziv-Ukelson, M., 2005. Alignment of metabolic pathways. *Bioinformatics* 21, 3401–3408.

Pollock, D., Taylor, W., and Goldman, N., 1999. Coevolving protein residues: maximum likelihood identification and relationship to structure. *Journal of Molecular Biology* 287(1), 187–198.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1992. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK.

Pu, S., Vlasblom, J., Emili, A., Greenblatt, J., and Wodak, S., 2007. Identifying functional modules in the physical interactome of saccharomyces cerevisiae. *Proteomics* 7, 944–960.

Rao, C. V., Kirby, J. R., and Arkin, A. P., 2004. Design and diversity in bacterial chemotaxis: a comparative study in *Escherichia coli* and *Bacillus subtilis*. *PLoS Biology* 2(2), E49.

Rives, A. and Galitski, T., 2003. Modular organization of cellular networks. *Proceedings of the National Academy of Sciences* 100, 1128–1133.

Robinson, D. M., Jones, D. T., Kishino, H., Goldman, N., and Thorne, J. L., 2003. Protein evolution with dependence among codons due to tertiary structure. *Molecular Biology and Evolution* 20(10), 1692–1704.

Rual, J.-F., Venkatesan, K., Hao, T., Hirozane-Kishikawa, T., Dricot, A., Li, N., Berriz, G. F., Gibbons, F. D., Dreze, M., Ayivi-Guedehoussou, N., and *et al.*, 2005. Towards a proteome-scale map of the human protein-protein interaction network. *Nature* 437(7062), 1173–1178.

Salwinski, L., Miller, C., Smith, A., Pettit, F., Bowie, J., and Eisenberg, D., 2004. The Database of Interacting Proteins. *Nucleic Acids Research* 32(Database issue), D449–D551.

Schellewald, C., 2005. *Convex mathematical programs for relational matching of object views*. Ph.D. thesis, Dept. of Mathematics and Computer Science, University of Mannheim.

Scott, J., Ideker, T., Karp, R., and Sharan, R., 2005. Efficient Algorithms for Detecting Signaling Pathways in Protein Interaction Networks. In *Proc. 9th Intl. Conf. on Research in Computational Molecular Biology*, 1–13.

Serres, M. and Riley, M., 2006. Genomics and Metabolism in *Escherichia coli*. In *The Prokaryotes*, chapter 1.10, 261–274. Springer, New York.

Shannon, P., Markiel, A., Ozier, O., Baliga, N., Wang, J., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T., 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research* 13(11), 2498–2504.

Sharan, R. and Ideker, T., 2006. Modeling cellular machinery through biological network comparison. *Nature Biotechnology* 24(4), 427–433.

Sharan, R., Ideker, T., Kelley, B., Shamir, R., and Karp, R., 2004. Identification of Protein Complexes by Comparative Analysis of Yeast and Bacterial Protein Interaction Data. In *Proc. 8th Intl. Conf. on Research in Computational Molecular Biology*, 282–289.

Sharan, R., Suthram, S., Kelley, R., Kuhn, T., McCuine, S., Uetz, P., Karp, R., and Ideker, T., 2005. Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences* 102(6), 1974–1979.

Sharan, R., Ulitsky, I., and Shamir, R., 2007. Network-based prediction of protein function. *Molecular Systems Biology* 3.

Shasha, D., Wang, J.-L., and Giugno, R., 2002. Algorithmics and Applications of Tree and Graph Searching. In *Proc. ACM Symp. on Principles of Database Systems*.

Siek, J., Lee, L., and Lumsdaine, A., 2002. *The Boost Graph Library: User guide and reference manual*. Addison-Wesley Professional.

Siepel, A. and Haussler, D., 2004. Combining phylogenetic and hidden Markov models in biosequence analysis. *Journal of Computational Biology* 11, 413–428.

Sonenshein, A. L., Hoch, J. A., and Losick, R., 2002. *Bacillus subtilis and its closest relatives*. ASM Press, Washington DC.

Spirin, V. and Mirny, L., 2003. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences* 100, 12123–12128.

Stelzl, U., Worm, U., Lalowski, M., Haenig, C., Brembeck, F. H., Goehler, H., Stroedicke, M., Zenkner, M., Schoenherr, A., Koeppen, S., and *et al.*, 2005. A Human Protein-Protein Interaction Network: A Resource for Annotating the Proteome. *Cell* 122(6), 957–968.

Stitson, M., Gammerman, A., Vapnik, V., Vovk, V., Watkins, C., and Weston, J., 1999. Support vector regression with ANOVA decomposition kernels. *Advances in kernel methods: Support vector learning*.

Storz, G. and Hengge-Aronis, R., 2000. *Bacterial Stress Responses*. ASM Press, Washington DC.

Stuart, J., Segal, E., Koller, D., and Kim, S., 2003. A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules. *Science* 302(5643), 249–255.

Sullivan, J. and Joyce, P., 2005. Model selection in phylogenetics. *Annual Reviews of Ecology, Evolution and Systematics* 36, 445–66.

Tan, K., Shlomi, T., Feizi, H., Ideker, T., and Sharan, R., 2007. Transcriptional regulation of protein complexes within and across species. *Proceedings of the National Academy of Sciences* 104, 1283–1288.

Tatusov, R. L., Koonin, E. V., and Lipman, D. J., 1997. A genomic perspective on protein families. *Science* 278(5338), 631–637. URL `http://www.sciencemag.org/cgi/content/abstract/278/5338/631`.

The Gene Ontology Consortium, 2000. Gene Ontology: tool for the unification of biology. *Nature Genetics* 25(1), 25–29.

Tian, Y., McEachin, R., Santos, C., States, D., and Patel, J., 2007. Saga: a subgraph matching tool for biological graphs. *Bioinformatics* 23, 232–239.

Uetz, P., Giot, L., Cagney, G., Mansfield, T. A., Judson, R. S., Knight, J. R., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P., and *et al.*, 2000. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature* 403(6770), 623–627.

von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., and Bork, P., 2002. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature* 417(6887), 399–403.

Walhout, A., Sordella, R., Lu, X., Hartley, J., Temple, G., Brasch, M., Thierry-Mieg, N., and Vidal, M., 2000. Protein interaction mapping in c. elegans using proteins involved in vulval development. *Science* 287, 116–122.

Waters, C. and Bassler, B., 2006. The *Vibrio harveyi* quorum-sensing system uses shared regulatory components to discriminate between multiple autoinducers. *Genes & Development* 20(19), 2754–67.

Weiss, Y., 1999. Segmentation using eigenvectors: a unifying view. In *Proc. IEEE Intl. Conf. on Computer Vision*, 975–982.

Wray, G. A., Hahn, M. W., Abouheif, E., Balhoff, J. P., Pizer, M., Rockman, M. V., and Romano, L. A., 2003. The evolution of transcriptional regulation in eukaryotes. *Molecular Biology and Evolution* 20(9), 1377–1419.

Xie, G., Keyhani, N. O., Bonner, C. A., and Jensen, R. A., 2003. Ancient origin of the tryptophan operon and the dynamics of evolutionary change. *Microbiology and Molecular Biology Reviews* 67(3), 303–42.

Yang, Q. and Sze, S.-H., 2007. Path matching and graph matching in biological networks. *Journal of Computational Biology* 14, 56–67.

Yang, Z., 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *Journal of Molecular Evolution* 39(3), 306–14.

Yang, Z., 1996. Maximum-likelihood models for combined analyses of multiple sequence data. *Journal of Molecular Evolution* 42, 587–96.

Yu, H., Luscombe, N. M., Lu, H. X., Zhu, X., Xia, Y., Han, J. D., Bertin, N., Chung, S., Vidal, M., and Gerstein, M., 2004. Annotation transfer between genomes: protein-protein interologs and protein-DNA regulogs. *Genome Research* 14(6), 1107–1118.

Zuckerkandl, E. and Pauling, L., 1965. Molecules as documents of evolutionary history. *Journal of Theoretical Biology* 8, 357–366.

Appendix A

Supplemental Text for Protein Network Comparison

We support the main text in Chapter 3 here by providing additional results and analyses. A Supplemental website (`http://www.cs.berkeley.edu/~nmani/M-and-S/`) has a freely available implementation of our Match-and-Split method. The Supplemental website also collects some conserved modules detected in our experiments, and associated functional descriptions and *predictions*.

Software for the previous methods we tested are from (`http://www.cs.tau.ac.il/~roded/networkblast.htm`) for NetworkBLAST and (`http://www.cs.purdue.edu/homes/koyuturk/mawish/`) for MaWISh.

A.1 Supplemental Tables

Method	Yeast-Human			Yeast-Fly			Yeast-Worm		
	#	Sens.	Spec.	#	Sens.	Spec.	#	Sens.	Spec.
Match-and-Split									
(p=1)	75	18.2	48.0	25	6.8	28.0	7	2.3	71.4
(p=2)	71	20.5	43.7	23	6.8	21.7	6	2.3	83.3
NetworkBLAST	923	28.0	18.9	158	7.6	46.8	9	1.5	88.9
MaWISh	169	17.4	46.8	48	6.1	35.4	8	2.3	75.0

Table A.1. Evaluation of output candidates from two-species comparisons using $sim(\cdot,\cdot)$ function based on criterion B. The results in other tables are based on criterion A. We use similar format as Table 3.1, but showing only module-level sensitivity and specificity expressed as rounded percentages. The "#" column shows the number of output modules.

Method	# candidates	% homogeneous		% similar
		Yeast	Human	
Match-and-Split				
(p=1)	80	100	83.8	42.5
(p=2)	72	98.6	80.6	40.3
NetworkBLAST	421	88.6	66.5	30.2
MaWISh	151	95.4	86.1	39.7

Table A.2. Percentage of output candidates from yeast-human comparison that are functionally homogeneous and similar (with respect to GO, as defined in Section 3.1.4). A higher percent (especially "% similar") suggests more candidates are likely to be conserved functional modules than spurious matches. The table thus provides informal specificity measures of the candidates using known GO annotations.

Method	# valid predictions	# total predictions
Match-and-Split		
(p=1)	295	462
(p=2)	297	459
NetworkBLAST	400	718
MaWISh	249	419

Table A.3. Validation of functional prediction of human proteins. The predictions result from annotation transfer on the output candidates from yeast-human comparison (see Section 3.2.4). The maximum number of predictions possible is 1882, as only 1882 of the 7355 proteins in the human network are sequence-similar to some protein in the yeast network (by the $sim(\cdot,\cdot)$ function).

A.2 Betweenness Clustering Heuristic

Our network comparison method incorporates a betweenness clustering heuristic to split large solutions (as seen in Section 3.1.3). We briefly describe this clustering procedure here. The procedure, taken from (Girvan and Newman, 2002), partitions a graph into highly-connected, smaller clusters based on iterative computations of an edge betweenness measure.

Consider all shortest paths between all node pairs in a graph, and assign the possibly multiple shortest paths between each node pair to equal weights summing to 1. The betweenness measure of an edge is then the sum of weights of these shortest paths that pass through the edge (see (Girvan and Newman, 2002) for other similar measures). The betweenness of all edges in a graph can be computed in $O(nm)$ time (Girvan and Newman, 2002), where n, m are the number of nodes, edges respectively in the graph.

The clustering procedure computes the betweenness measure of all edges in the graph, removes the edge with the maximum betweenness, and repeats these two steps iteratively on the reduced graph. We stop the procedure when the size of the largest connected component in the reduced graph becomes at most $n_{\max}$ (the same threshold as in Section 3.1.3). Each connected component of the final reduced graph defines a cluster of the input graph. We use the Boost Graph Library's (Siek *et al.*, 2002) implementation of this procedure.

The above procedure recomputes the edge betweenness measures after every edge removal, so it performs m iterations in the worst case with each taking $O(nm)$ time. A quicker version could recompute only when the number of connected components in the reduced graph increases by one. This version is a work-around to cluster large graphs over thousands of nodes, and is used in one of our experiments (see Table 3.5).

A.3 Statistical Significance - Analytical Bound

We discuss the statistical significance of our simple scoring measure here. Specifically, we upper bound the P-value of the score of a candidate under a null model that randomizes

the input data. Recall from Section 3.1.3 that the score of a candidate conserved module $S \subseteq G, T \subseteq H$, where G, H are the input protein networks, is the number of pairs of similar length-p paths between them.

We first specify how the null model randomizes the edges of G to obtain a random graph $\widetilde{G}$ over the same set of nodes (the case of $H, \widetilde{H}$ is similar). Independently assign an edge between every node pair u, v in $\widetilde{G}$ with probability $\frac{d(u)d(v)}{2m_G}$, where $d(x)$ refers to the degree of node x and m_G the number of edges in G (see (Berg and Lassig, 2004)). The null model also randomizes the node similarity function by independently setting $sim(u, v)$ for every node pair u, v in $\widetilde{G}, \widetilde{H}$ to true with probability p_r. Here p_r is the fraction of node pairs in G, H that are similar by the $sim(\cdot, \cdot)$ function.

Consider the candidate $S \subseteq G, T \subseteq H$, and let $\widetilde{S} \subseteq \widetilde{G}$ be the induced subgraph of $\widetilde{G}$ over the same set of nodes as S (similarly define $\widetilde{T} \subseteq \widetilde{H}$). The P-value of this candidate S, T is then obtained by comparing its score a to the score X of the random counterpart $\widetilde{S}, \widetilde{T}$. To simplify calculations, we bound the P-value of score a by $E[X]/a$, where $E[.]$ stands for expectation. We compute it as $E[X] = E[Q_1]E[Q_2]P_s$. The term $P_s \leq 2p_r^{p+1}$ is the probability for a pair of length-p paths from $\widetilde{S}, \widetilde{T}$ to be similar, with the factor 2 accounting for both orientations of a path. The term $E[Q_1]$ is the expected number of length-p paths in $\widetilde{S}$ and we upper bound it next (the bound for $E[Q_2]$, the expected number of paths in $\widetilde{T}$, is similar). Let the nodes in $\widetilde{S}$ be numbered $1, 2, \ldots, n$ and let $d(x), m_G$ of G be defined as above. Then,

$$\begin{aligned}
E[Q_1] &= \frac{1}{2} \sum_{1 \leq i_1 \neq i_2 \ldots \neq i_{p+1} \leq n} P[(i_1, i_2) \text{ is edge}] P[(i_2, i_3) \text{ is edge}] \ldots P[(i_p, i_{p+1}) \text{ is edge}] \\
&= \frac{1}{2} \sum_{1 \leq i_1 \neq i_2 \ldots \neq i_{p+1} \leq n} \frac{d(i_1)d(i_2)}{2m_G} \frac{d(i_2)d(i_3)}{2m_G} \cdots \frac{d(i_p)d(i_{p+1})}{2m_G} \\
&\leq \frac{(p+1)!}{2\,(2m_G)^p} \sum_{1 \leq i_1 < i_2 \ldots < i_{p+1} \leq n} d(i_1)^2 d(i_2)^2 d(i_3)^2 \ldots d(i_{p+1})^2
\end{aligned}$$

We compute the summation above, denoted $S[p+1, n]$, in $O(np)$ time using the recurrence $S[l, k] = S[l, k-1] + d(k)^2\ S[l-1, k-1]$ (similar to the recurrence in (Stitson *et al.*, 1999)).

To incorporate the edge reliabilities of noisy protein interactions, we use the expected

score $E[a]$ instead of the score a in the P-value bound above. We can readily compute $E[a]$ because each pair of similar paths between S, T, which contributes one to the score a, contributes the product of their edge reliabilities to $E[a]$.

Appendix B

Supplemental Text for Pathway Phylogeny Estimation

We support the main text in Chapter 4 by providing details on the input data and related analyses.

B.1 Input Data for Microbial Pathways

To build phylogenetic profiles of genes in pathways, we use 9 genes (11 interactions) in glycolysis, 12 genes (16 interactions) in the citric acid cycle, 13 genes (18 interactions) in chemotaxis, and 28 genes (28 interactions) in quorum sensing. The genes and interactions in a pathway are shown alongside the pathway phylogeny in the figures of Chapter 4. The exact sequence identifiers of these pathway genes are available too (M. Narayanan, AH. Singh, RM. Karp, manuscript under preparation). The genes in a pathway are chosen after extensive literature search for experimental evidence (genetic, biochemical, or high-throughput expression data) linking each gene to the pathway in model organisms. We use the model organisms *E. coli* for glycolysis and citric acid cycle, *B. subtilis* and *E. coli* for chemotaxis, and *V. harveyi*, *B. subtilis*, and *A. tumefaciens* for quorum sensing.

DNA and amino acid sequences for all genes were retrieved from the MicrobesOn-

line database (Alm *et al.*, 2005) on 3 Aug 2006 and their orthologs identified by a 3-way bi-directional best hit algorithm as previously described (Tatusov *et al.*, 1997), with the additional constraint that the sequence alignment coverage had to be at least 75% of the length of both genes. For phylogenetic profiles, a species was marked as having a gene if it had at least one ortholog, but possibly also multiple paralogs, of the gene. Ortholog sets for transcriptional regulators and histidine kinases, which are known to have highly conserved domains (Aravind *et al.*, 1999; Grebe and Stock, 1999), were manually curated to remove spurious hits by examining the phylogenetic tree for each ortholog set. Nucleotide and amino acid alignments were performed using Muscle (Edgar, 2004) with maxiters=3 and diags=1 (paralogs were discarded before the alignment step). Phylogenetic trees were built from aligned amino acid sequences using PHYML (Guindon and Gascuel, 2003) with default optimization parameters and 100 bootstrap replicates.